

Ref G

02P12391



19 BUNDESREPUBLIK
DEUTSCHLAND



DEUTSCHES
PATENTAMT

12 Übersetzung der
europäischen Patentschrift

87 EP 0 530 791 B 1

10 DE 692 21 046 T 2

51 Int. Cl. 6: 31
G 05 B 19/418
G 05 B 19/414
H 03 M 7/08

- 21 Deutsches Aktenzeichen: 692 21 046.6
86 Europäisches Aktenzeichen: 92 115 049.6
86 Europäischer Anmeldetag: 2. 9. 92
87 Erstveröffentlichung durch das EPA: 10. 3. 93
97 Veröffentlichungstag
der Patenterteilung beim EPA: 23. 7. 97
47 Veröffentlichungstag im Patentblatt: 5. 3. 98

- 30 Unionspriorität:
222802/91 03. 09. 91 JP
- 73 Patentinhaber:
Mitsubishi Denki K.K., Tokio/Tokyo, JP
- 74 Vertreter:
Meissner, Bolte & Partner, 80538 München
- 84 Benannte Vertragsstaaten:
DE, FR, GB

- 72 Erfinder:
Hirai, Hayao, c/o Mitsubishi Denki K. K., Higashi-ku,
Nagoya-shi, Aichi-ken, JP

- 54 Datenübertragungsverfahren

Anmerkung: Innerhalb von neun Monaten nach der Bekanntmachung des Hinweises auf die Erteilung des europäischen Patents kann jedermann beim Europäischen Patentamt gegen das erteilte europäische Patent Einspruch einlegen. Der Einspruch ist schriftlich einzureichen und zu begründen. Er gilt erst als eingelegt, wenn die Einspruchsgebühr entrichtet worden ist (Art. 99 (1) Europäisches Patentübereinkommen).

Die Übersetzung ist gemäß Artikel II § 3 Abs. 1 IntPatÜG 1991 vom Patentinhaber eingereicht worden. Sie wurde vom Deutschen Patentamt inhaltlich nicht geprüft.

DE 692 21 046 T 2

DE 692 21 046 T 2

DATENÜBERTRAGUNGSVERFAHREN

Die Erfindung betrifft serielle Datenübertragungsverfahren
5 zwischen einer Hauptsteuereinheit und verschiedenen Terminals
innerhalb eines automatisierten Fabriksteuersystems bzw. FA-
Steuersystems. Beispielsweise betrifft die Erfindung ein seri-
elles Datenübertragungsverfahren zwischen der Hauptsteuerein-
heit und den Terminals innerhalb eines numerisch gesteuerten
10 Werkzeugmaschinen-systems, wie etwa eines Displaysystems, eines
Servosystems, eines Spindelsystems usw..

Die Datenübertragung innerhalb eines automatisierten Fabriksy-
stems erfolgt entweder (1) durch eine Busverbindung oder eine
15 serielle Übertragung unter Nutzung von Zweitor-RAMs, bei denen
das Einschreiben und Auslesen in zwei Richtungen (d.h. an zwei
Toren) erfolgen kann, oder (2) durch selbständige Speicher,
die an der Hauptsteuereinheit bzw. den Terminals angeordnet
sind, wobei die Übertragung und der Empfang der Daten oder der
20 Anforderungssignale zu und von den Speichern über eine sequen-
tielle Übertragungsleitung erfolgt. Als nächstes wird das
zweitgenannte Verfahren der Datenübertragung beschrieben, wo-
bei einzelne Lichtwellenleiter die Speicher der Hauptsteuer-
einheit und der verschiedenen Terminals miteinander verbinden.

25 Fig. 18 ist ein Blockbild, das die herkömmliche Organisation
eines automatisierten Fabriksystems zeigt. Die Hauptsteuerein-
heit 1 besteht aus einer Steuereinheit 101, einem Speicher 102
und einer Eingabe/Ausgabe-Schnittstelle 103. Die Terminals um-
fassen ein Displaysystem 2, Servosysteme 3 und Spindelsysteme
30 4, die mit der Hauptsteuereinheit 1 über jeweilige einzelne
Lichtleiterkabel 5, 6 verbunden sind. Jedes System der Termi-
nals 2, 3 und 4 umfaßt eine Steuereinheit 201, 301, 401, einen

5 Speicher 202, 302, 402 und eine Eingabe/Ausgabe-Schnittstelle 203, 303, 403. Außerdem weist das Displaysystem 2 ein Display 204 auf. Jedes der Servosysteme 3 weist einen Servomotor 304 auf. Jedes der Spindelsysteme 4 weist einen Spindelmotor 404 auf.

10 Die Datenübertragung über die Lichtleiterkabel 5 und 6 zwischen der Hauptsteuereinheit 1 und den Terminals 2, 3 und 4 erfolgt in zwei Richtungen. Die Daten und die Parameter, die für die Steuerung der numerisch gesteuerten Werkzeugmaschinen erforderlich sind und die von der Steuereinheit 101 der Hauptsteuereinheit 1 erzeugt werden, werden zuerst in dem Speicher 102 gespeichert. Die Daten, die vorübergehend in dem Speicher 102 gespeichert werden, und die Parameter, die periodisch aktualisiert werden, werden über die jeweiligen Lichtleiterkabel 15 5 und 6 zu der jeweiligen Eingabe/Ausgabe-Schnittstelle 203, 303 und 403 des Displaysystems 2, der Servosysteme 3 und der Spindelsysteme 4 übertragen und darin gespeichert. Die Steuereinheit 201, 301, 401 jedes Systems der jeweiligen Terminals führt entsprechende Funktionen (das Anzeigen von Information auf dem Display 204 im Fall des Displaysystems 2 und die Steuerung des Servomotors 304 oder des Spindelmotors 404 im Fall der Servosysteme 3 oder der Spindelsysteme 4) auf der Basis dieser Daten und Parameter aus, die in ihren jeweiligen Speichern abgelegt sind. 20 25

Andererseits werden die Daten, die an den Terminals, wie etwa dem Displaysystem 2, den Servosystemen 3 und den Spindelsystemen 4 detektiert oder eingegeben werden, in vorbestimmte Datenformate durch die jeweilige Steuereinheit 201, 301, 401 konvertiert, und wenn sie vorübergehend in dem jeweiligen Speicher 202, 302 und 402 gespeichert worden sind, werden sie nach Bedarf zu dem Speicher 102 der Hauptsteuereinheit 1 über die jeweilige Eingabe/Ausgabe-Schnittstelle 203, 303, 403, das 30 Lichtleiterkabel 5, 6 und die Eingabe/Ausgabe-Schnittstelle 103 der Steuereinheit 101 übertragen. Die Datenübertragung er- 35

folgt beispielsweise gemäß den folgenden Industriestandards: RS232C, RS363, RS404, RS491, RS422, die von der EIA (Electronics Industries Association) festgelegt sind. Die Übertragungsgeschwindigkeit ist jedoch durch das Leistungsvermögen der für die Übertragung verwendeten Einrichtungen be-
5 grenzt. Die maximale Übertragungsgeschwindigkeit der seriellen Datenübertragung mittels RS491 ist ferner 19200 bit/s. Die maximale Übertragungsgeschwindigkeit mittels RS422 ist 2 Mbit/s.

10 Die von der Hauptsteuereinheit 1 zu den Servosystemen 3 oder den Spindelsystemen 4 übertragenen Daten und Signale umfassen Positionsdaten, Geschwindigkeitsdaten, Steuerdaten und Verstärkungssignale. Die von den Servosystemen 3 oder den Spindelsystemen 4 zu der Hauptsteuereinheit 1 übertragenen Rückführsignale umfassen Positionsrückführdaten, Geschwindigkeitsrückführdaten, Lastrückführdaten und Warnsignale. Außerdem
15 werden nach Bedarf die Parameter für mehr als zwanzig Posten für jede der Servo- und der Hauptspindeln von der Hauptsteuereinheit 1 zu den Servosystemen 3 oder den Spindelsystemen 4 übertragen.
20

Unter den obigen Daten umfassen die an den Servosystemen 3 und den Spindelsystemen 4 empfangenen Daten (die Positionsdaten, die Geschwindigkeitsdaten, die Steuerdaten, die Verstärkungssignale usw.) und die von dort übertragenen Daten (die Positionsrückführdaten, die Geschwindigkeitsrückführdaten, die Lastrückführdaten, die Warnsignale usw.) Daten für eine Vielzahl von Achsen der Servosysteme 3 oder der Spindelsysteme 4 innerhalb eines vorbestimmten Steuerzeitintervalls.
25

30 Fig. 19 ist ein Diagramm, das die herkömmlichen Datenformate der Informationen zeigt, die zwischen der Hauptsteuereinheit und den Terminals übertragen werden. Die Daten und Signale werden unter Anwendung von ASCII-Codes (American Standard Code for Information Interchange) übertragen. Die Information oder
35 das Signal, das an den Servosystemen 3 oder den Spindelsyste-

men 4 empfangen wird und jeweils aus einem Vorsatzelement, einem Datenelement und einem Endelement besteht, weist folgendes auf:

5 Positionsdaten (a), die in 12 Bytes untergebracht sind,
Geschwindigkeitsdaten (b) in 10 Bytes,
erste Steuerdaten (c) in 5 Bytes,
zweite Steuerdaten (d) in 5 Bytes,
das erste Verstärkungssignal (e) in 11 Bytes, und
das zweite Verstärkungssignal (f) in 9 Bytes.

10 Die Anzahl Bytes ist die Summe der Anzahl von Bytes des Vorsatzelements, des Datenelements und des Endelements. Die Länge jedes Elements in Bytes ist in jeweiligen Skalendiagrammen in Fig. 19 dargestellt. Die an den Servosystemen 3 oder den Spindel-
15 systemen 4 empfangenen Daten entsprechen somit insgesamt 52 Bytes.

Die Daten und Signale, die von den Servosystemen 3 oder den Spindel-
20 systemen 4 zu der Hauptsteuereinheit 1 übertragen werden, umfassen folgendes:

die Positionsrückführdaten (g), die in 12 Bytes angeordnet sind,
die Geschwindigkeitsrückführdaten (h) in 10 Bytes,
die Lastrückführdaten (i) in 8 Bytes und
25 das Warnsignal (j) in 7 Bytes.

Jedes dieser Daten oder Signale besteht ebenfalls aus einem Vorsatzelement, einem Datenelement und einem Endelement. Die Gesamtzahl der Bytes von übertragenen Daten beträgt 37 Bytes.

30 Fig. 20 ist ein Diagramm, das ein herkömmliches Beispiel der Daten zeigt, die innerhalb eines vorbestimmten Intervalls einer Steuerzeit (IT) empfangen und übertragen werden. Die Servosysteme 3 und die Spindelsysteme 4 weisen eine Vielzahl von
35 Achsen auf. Daher umfaßt die empfangene und die übertragene Information innerhalb eines vorbestimmten Steuerzeitintervalls

(IT) Daten für n Achsen, wobei n die Anzahl von gesteuerten Achsen ist. Um den Start und das Ende der empfangenen und der übertragenen Daten zu markieren, werden die Startdaten aus 2 Bytes und die Enddaten aus 1 Byte den empfangenen bzw. den übertragenen Daten hinzugefügt. Die gesamten Daten einschließlich der empfangenen und der übertragenen Daten müssen innerhalb eines einzigen IT (vorbestimmten Steuerzeitintervalls) übertragen werden. Fig. 20 zeigt den Fall, in dem die gesamte Übertragungszeit für die empfangenen und die übertragenen Daten genau gleich einem IT (vorbestimmten Steuerzeitintervall) ist. Im allgemeinen kann die Gesamtübertragungszeit weniger als ein IT sein, so daß am Ende jedes IT ein Spielraum verbleibt. Die Gesamtbytes, die innerhalb eines ITs zu übertragen sind, sind folgende:

$$(2 + 1) + 52 n + (2 + 1) + 37 n \text{ Bytes.}$$

Wenn für die Übertragung von Bytes der EIA-Standard RS404 angewandt wird, sind für die Übertragung eines einzelnen Bytes 10 Bits erforderlich. Außerdem ist gemäß dem EIA-Standard RS422 die Datenübertragungsgeschwindigkeit 2 Mbit/s. Wenn man also davon ausgeht, daß die Länge eines IT gleich 2 ms ist, dann ist die Anzahl n von Achsen, die innerhalb eines IT gesteuert werden können:

$$\begin{aligned} & (2 + 1) \times 10 + 52n \times 10 + (2 + 1) \times 10 + 37n \times 10 \\ & = 2\,000\,000 \times 0,002 \end{aligned} \quad (1).$$

Unter Vereinfachung der obigen Gleichung erhält man:

$$60 + 890n = 4\,000 \quad (1')$$

und somit ungefähr

$$n = 4,43 \quad (1'').$$

Daraus kann also gefolgert werden, daß bis zu vier Achsen gesteuert werden können.

5 Mit dem obigen herkömmlichen Datenübertragungsverfahren ist es also möglich, bis zu vier Spindeln oder Achsen der Servosysteme 3 oder der Spindelssysteme 4 zu steuern. Im Fall der numerisch gesteuerten Werkzeugmaschinen bedeutet dies, daß das herkömmliche Verfahren bei Einspindel-Einschlitten-Drehmaschinen oder Einspindelkopf-Bearbeitungszentren anwendbar ist. Das
10 herkömmliche Datenübertragungsverfahren kann jedoch nicht angewandt werden, um Werkzeugmaschinen mit Hilfsachsen oder Mehrsystem-Drehmaschinen oder Bearbeitungszentren mit einer Vielzahl von Hauptspindeln und Schlitten zu steuern.

15 Seit einiger Zeit steigt die Zahl von gesteuerten Achsen von numerisch gesteuerten Werkzeugmaschinen wegen der Forderung nach einem höheren Automatisierungsniveau mit Multifunktionalität und auch wegen der Forderung nach höherer Leistungsfähigkeit und der Konvertierung der PLC-Achsen (PLC = programmierbare logische Steuerung) in numerisch gesteuerte Achsen. Außerdem erhält zusätzlich zu der herkömmlichen seriellen Datenübertragung die Datenübertragung über ein einziges Lichtleiterkabel hohe Priorität, und zwar gemeinsam mit der Verringerung der Zahl von Montageschritten der Werkzeugmaschine, der
20 Herabsetzung der Häufigkeit des Auftretens von Störungen und der Verbesserung der Abschirmung gegenüber äußeren Störungen.

Unter diesen Umständen kann das herkömmliche serielle Datenübertragungsverfahren mit maximal vier Achsen die obigen Forderungen nicht erfüllen. Die Erhöhung der Zahl von steuerbaren
30 Spindeln oder Achsen ist daher ein dringendes Anliegen.

Es ist somit eine Aufgabe der Erfindung, ein Datenübertragungsverfahren anzugeben, das den herkömmlichen Standards der
35 Datenübertragung genügt, wobei die Menge an übertragener In-

formation und somit die Zahl der gesteuerten Achsen erhöht werden kann.

5 Diese Aufgabe wird nach der Erfindung durch ein serielles Datenübertragungsverfahren gemäß der Definition in Anspruch 1 gelöst.

10 Die Konvertierung erfolgt bevorzugt mittels einer Softwarelogik, die mit einem Programm implementiert wird, mittels eines Tabellennachschlagverfahrens auf der Basis einer Konvertierungstabelle, die die Korrelation von Dezimalzahlen zu den
15 Zahlencodes auf der Basis n herstellt, oder mittels eines Tabellennachschlagverfahrens mit schrittweiser Konvertierung. Die Tabelle für schrittweise Konvertierung kann eine Korrelation von Dezimalzahlen, die auf ein nächstliegendes Vielfaches einer ganzen Zahl gerundet sind, zu den Zahlencodes auf der
20 Basis n herstellen, die den gerundeten Dezimalzahlen, dividiert durch die ganze Zahl, entsprechen, wobei die ganze Zahl gleich oder größer als 2 ist.

25 Das Betriebsverfahren der Erfindung selbst ergibt sich am besten aus der nachstehenden genauen Beschreibung von Ausführungsformen in Verbindung mit den beigefügten Zeichnungen; in den Zeichnungen zeigen:

- 30 Fig. 1 ein Flußdiagramm, das den Ablauf für die Datenübertragung von der Hauptsteuereinheit 1 zu den Servosystemen 3 und den Spindelsystemen 4 der Fig. 18 zeigt;
Fig. 2 ein Flußdiagramm, das den Ablauf für den Datenempfang an den Servosystemen 3 und den Spindelsystemen 4 der Fig. 18 zeigt;
35 Fig. 3 ein Flußdiagramm, das den Ablauf für die Datenübertragung von den Servosystemen 3 und den Spindelsystemen 4 zu der Hauptsteuereinheit 1 von Fig. 18 zeigt;

- Fig. 4 ein Flußdiagramm, das den Ablauf für den Datenempfang an der Hauptsteuereinheit 1 von Fig. 18 zeigt;
- Fig. 5 eine Tabelle, die die Konvertierung der im Code mit der Basiszahl 10 dargestellten Zahlen zu entsprechenden Codes mit der Basiszahl 36 zeigt, wobei die ganz rechte Spalte die sieben höchstwertigen Stellen der Zahlen auf Basis 10 und die Kopfzeile die niederwertigste Stelle davon zeigt;
- Fig. 6 eine Tabelle, die die Konvertierung der im Code mit der Basiszahl 36 dargestellten Zahlen in entsprechende Dezimalzahlen zeigt, wobei die ganz rechte Spalte die fünf höchstwertigen Stellen der Codezahlen mit Basiszahl 36 und die Kopfzeile die niederwertigste Stelle davon zeigt;
- Fig. 7 eine schrittweise Konvertierungstabelle zur Konvertierung der sechsstelligen Zahlen des Codes mit der Basiszahl 10 in Codes mit der Basiszahl 36 in Zweierschritten, wobei die ganz rechte Spalte die fünf höchstwertigen Stellen der Zahlen mit Basiszahl 10 und die Kopfzeile die niederwertigste Stelle davon zeigt;
- Fig. 8 eine umgekehrt gestufte Konvertierungstabelle zur Wiederherstellung der gestuften Codes mit der Basiszahl 36, wie sie unter Anwendung der Tabelle von Fig. 7 erhalten sind, in entsprechende sechsstellige Dezimalzahlen in Zweierschritten, wobei die ganz rechte Spalte die zwei höchstwertigen Stellen der Codezahlen mit Basiszahl 36 und die Kopfzeile die niederwertigsten Stellen davon zeigt;
- Fig. 9 eine Tabelle zum Vergleich der fest vorgegebenen Blockdatenübertragungsformate entsprechend der Zahlencodierung mit Basiszahl 10 (linke Spalte) und der Codierungen mit Basiszahl 36 (rechte Spalte) zwischen der Hauptsteuereinheit 1 und den Servosystemen 3 oder den Spindelsystemen 4 von Fig. 18;

Fig. 10 eine Konvertierungstabelle von dezimalcodierten Zahlen mit Basiszahl 36 zu Codeziffern mit Basiszahl 36;

5 Fig. 11 eine Konvertierungstabelle von den Codeziffern mit Basiszahl 36 zu dezimalcodierten Zahlen mit Basiszahl 36;

Fig. 12 ein Diagramm, das die Datenformate für die empfangenen und übertragenen Daten zeigt, wobei die internen Vorsatzelemente eliminiert sind;

10 Fig. 13 ein Diagramm, das die Datenformate der empfangenen und übertragenen Daten zeigt, wobei die internen Vorsatzelemente und Stoppelemente eliminiert sind;

Fig. 14 ein Diagramm, das die Datenformate der empfangenen und übertragenen Daten ähnlich Fig. 13 zeigt, wobei die Datenwörter weiter komprimiert sind;

15 Fig. 15 ein Diagramm, das das Format der Daten zeigt, die innerhalb eines vorbestimmten Steuerzeitintervalls empfangen und übertragen werden, wobei die Datenformate von Fig. 14 angewandt werden;

20 Fig. 16 ein Diagramm, das ein System zeigt, das um einen FA-Mastercomputer mit untergeordneten Zellensteuerungen zentriert ist, mit denen die verschiedenen untergeordneten Systeme gekoppelt sind;

25 Fig. 17 ein Diagramm, das ein der Fig. 16 ähnliches System zeigt, wobei jedoch die Zellensteuerungen weggelassen sind;

Fig. 18 ein Blockbild, das die herkömmliche Organisation eines automatisierten Fabriksystems zeigt;

30 Fig. 19 ein Diagramm, das die herkömmlichen Datenformate der Information zeigt, die zwischen der Hauptsteuereinheit und den Terminals übertragen wird; und

Fig. 20 ein Diagramm eines Beispiels der herkömmlichen Daten, die innerhalb eines vorbestimmten Steuerzeitintervalls (IT) empfangen und übertragen werden.

In den Zeichnungen sind gleiche oder entsprechende Teile oder Bereiche mit den gleichen Bezugszeichen versehen.

5 Unter Bezugnahme auf die beigegeführten Zeichnungen werden die bevorzugten Ausführungsformen der Erfindung beschrieben, wobei Fig. 18 als Beispiel eines automatisierten Fabriksystems dient.

10 Fig. 9 ist eine Tabelle, die die festgelegten Blockdatenübertragungsformate entsprechend der Zahlencodierung auf Basis 10 (linke Spalte) und der Zahlencodierung auf Basis 36 (rechte Spalte) zwischen der Hauptsteuereinheit 1 und den Servosystemen 3 oder den Spindelsystemen 4 von Fig. 18 vergleicht. Der Maximalwert der Positionsdaten (a) ist in einer acht Stellen
15 oder Ziffern aufweisenden Zahl 99999999 auf Basis 10 dargestellt. Um also die Positionsdaten in Dezimalzahlen (auf Basis 10) darzustellen, sind acht (8) Bytes erforderlich. Durch aufeinanderfolgende Division der maximalen Daten 99999999 durch 36 und Bilden der aufeinanderfolgenden Reste kann die dezimalcodierte hexatrigintale (Basis 36) Zahl erhalten werden. So
20 kann der maximale Datenwert dezimalcodiert hexatrigintal (Basis 36) in sechs Blöcken dargestellt werden: [1] [23] [19] [12] [17] [27] (dezimalcodierte Basis 36). Das gleiche wird als 1NJCHR im Code auf Basis 36 dargestellt, wobei die Zahlen
25 10 bis 35 durch Buchstaben A bis Z dargestellt sind, wie Fig. 10 zeigt. So können die durch Basis 36 dargestellten Positionsdaten in sechs Bytes komprimiert werden.

Gleichermaßen ist der Maximalwert der Geschwindigkeitsdaten
30 (b) durch eine sechsstellige Dezimalzahl 999999 (Basis 10) dargestellt. Um also die Geschwindigkeitsdaten in Dezimalform darzustellen, sind sechs (6) Bytes erforderlich. Das gleiche Maximum kann dezimalcodiert hexatrigintal (Basis 36) in vier Blöcken dargestellt werden: [21] [15] [21] [27] (dezimalco-
35 diert, Basis 36). Das gleiche Maximum ist als LFLR im Code auf

Basis 36 dargestellt, der somit in vier Bytes zusammengefaßt ist.

Die Maximalwerte der ersten Steuerdaten (c) und der zweiten Steuerdaten (d) sind jeweils durch eine einstellige Dezimalzahl 9 (Basis 10) dargestellt. Die beiden Datenwörter, die jeweils von 0 bis 9 (Basis 10) reichen, werden daher zu einem einzigen Datenwort mit einem gemeinsamen Vorsatz kombiniert. Der Wert der resultierenden kombinierten Steuerdaten liegt zwischen 0 und 19 (Basis 10). Dann kann der Maximalwert der kombinierten Daten dezimalcodiert hexatrigintal (Basis 36) in einem Block dargestellt werden: [19] (dezimalcodiert Basis 36). Das gleiche Maximum wird durch ein einzelnes Byte im Code auf Basis 36 dargestellt: J (Basis 36).

Der Maximalwert des ersten Verstärkungssignale (e) ist durch eine fünfstellige Dezimalzahl 99999 (Basis 10) dargestellt. Um das erste Verstärkungssignal dezimal darzustellen, sind fünf (5) Bytes notwendig. Der Maximalwert kann dezimalcodiert hexatrigintal (Basis 36) in vier Blöcken dargestellt werden: [2] [5] [5] [27] (dezimalcodiert Basis 36). Das gleiche Maximum ist als 255R (Basis 36) in Code auf Basis 36 dargestellt, das daher in vier Bytes zusammengefaßt werden kann. Ebenso ist der Maximalwert des zweiten Verstärkungssignals (f) durch eine dreistellige Dezimalzahl 999 (Basis 10) dargestellt. Um das zweite Verstärkungssignal dezimal darzustellen, sind drei (3) Bytes notwendig. Der Maximalwert kann dezimalcodiert hexatrigintal (Basis 36) in zwei Blöcken dargestellt werden: [27] [27] (dezimalcodiert Basis 36). Das gleiche Maximum wird in Code auf Basis 36 als RR (Basis 36) dargestellt und kann in zwei Bytes zusammengefaßt werden.

Ferner ist der Maximalwert der Positionsrückführdaten (g) durch eine achtstellige Dezimalzahl 99999999 (Basis 10) dargestellt. Um die Positionsrückführdaten dezimal darzustellen, sind acht (8) Bytes erforderlich. Der Maximalwert kann dezi-

malcodiert hexatrigintal (Basis 36) in sechs Blöcken dargestellt werden: [1] [23] [29] [12] [17] [27] (dezimalcodiert Basis 36). Das gleiche Maximum wird im Code auf Basis 36 als 1NJCHR (Basis 36) dargestellt, das in sechs Bytes zusammengefaßt werden kann.

Gleichermaßen wird der Maximalwert der Geschwindigkeitsrückführdaten (h) durch eine sechsstellige Dezimalzahl 999999 (Basis 10) dargestellt. Um die Geschwindigkeitsrückführdaten dezimal darzustellen, sind sechs (6) Bytes notwendig. Der Maximalwert kann dezimalcodiert hexatrigintal (Basis 36) in vier Blöcken dargestellt werden: [21] [15] [21] [27] (dezimalcodiert Basis 36). Das gleiche Maximum wird in Code auf Basis 36 als LFLR (Basis 36) dargestellt und kann in vier Bytes zusammengefaßt werden.

Der Maximalwert der Lastrückführdaten (i) ist durch eine vierstellige Dezimalzahl 9999 (Basis 10) dargestellt. Um also die Lastrückführdaten dezimal darzustellen, sind vier (4) Bytes notwendig. Der Maximalwert kann dezimalcodiert hexatrigintal (Basis 36) in drei Blöcken dargestellt werden: [7] [25] [27] (dezimalcodiert Basis 36). Das gleiche Maximum wird in Code auf Basis 36 als 7PT (Basis 36) dargestellt und kann in drei Bytes zusammengefaßt werden.

Der Maximalwert des Warnsignals (i) wird durch eine dreistellige Dezimalzahl 999 (Basis 10) dargestellt. Um also das Warnsignal dezimal darzustellen, sind drei (3) Bytes notwendig. Der Maximalwert kann dezimalcodiert hexatrigintal (Basis 36) in zwei Blöcken dargestellt werden: [28] [27] (dezimalcodiert Basis 36). Das gleiche Maximum wird in Code auf Basis 36 als RR (Basis 36) dargestellt und kann in zwei Bytes zusammengefaßt werden.

Unter Anwendung des Codes auf der Basis 36 können also die an den Servosystemen 3 oder den Spindelsystemen 4 (siehe Fig. 18)

empfangenen Daten in 17 Bytes gegenüber 24 Bytes im Fall der Basis 10 komprimiert werden. Ferner werden das Vorsatzelement und das Endelement für die Steuerdaten gemeinsam genutzt. Somit kann die Gesamtzahl von Bytes von empfangenen Daten einschließlich der Vorsatz- und Endelemente für jeweilige Datenwörter, die sich im Fall der Basis 10 auf 52 Bytes summierten (siehe Fig. 19), durch Anwendung des Codes auf der Basis 36 zu 41 Bytes komprimiert werden. Ebenso werden die von den Servosystemen 3 oder den Spindelsystemen 4 übertragenen Daten in Basis 36 zu 15 Bytes komprimiert gegenüber 21 Bytes im Fall von Basis 10. Somit wird die Gesamtzahl von Bytes von übertragenen Daten einschließlich der Vorsatz- und der Endelemente für jeweilige Datenwörter auf 31 Bytes gegenüber 37 für Basis 10 verringert (siehe Fig. 19). Wenn also die Information nach der Codierung in Basis 36 übertragen wird, ist die Anzahl n der Spindeln oder Achsen, die innerhalb eines IT (eines vorbestimmten Steuerzeitintervalls) gesteuert werden können, folgende:

$$(2 + 1) \times 10 + 41n \times 10 + (2 + 1) \times 10 + 31n \times 10 = 2\,000\,000 \times 0,002 \quad (2).$$

Durch Vereinfachung der obigen Gleichung erhält man:

$$60 + 720n = 4000 \quad (2')$$

und somit ungefähr

$$n = 5,42 \quad (2'').$$

Es können also bis zu fünf Achsen gesteuert werden.

In der vorstehenden Erörterung hat die Information immer noch ein festgelegtes Blockdatenformat mit einem Vorsatz für jedes einzelne Datenwort. Fig. 12 ist ein Diagramm der vorliegenden Ausführungsform und zeigt die Datenformate für die empfangenen und die übertragenen Daten, wobei die internen Vorsatzelemente

entfallen. Dabei sind im Fall der Daten, die an den Servosystemen 3 oder den Spindelsystemen 4 von Fig. 18 empfangen werden, die Vorsatzelemente mit Ausnahme am Kopf der Positionsdaten eliminiert. Die empfangenen Daten werden dadurch um 16 Bytes komprimiert. Im Fall der von den Servosystemen 3 oder den Spindelsystemen 4 übertragenen Daten sind die Vorsätze mit Ausnahme am Kopf der Positionsrückführdaten eliminiert. Die übertragenen Daten werden dadurch um 9 Bytes komprimiert.

Dann ist die Anzahl n von Spindeln oder Achsen, die innerhalb eines IT (eines vorbestimmten Steuerzeitintervalls) gesteuert werden können:

$$(2 + 1) \times 10 + 25n \times 10 + (2 + 1) \times 10 + 22n \times 10 = 2\,000\,000 \times 0,002 \quad (3).$$

Durch Vereinfachung der vorstehenden Gleichung erhält man:

$$60 + 470n = 4000 \quad (3')$$

und damit ungefähr

$$n = 8,38 \quad (3'').$$

Somit können bis zu acht Achsen gesteuert werden.

Fig. 13 ist ein Diagramm einer anderen Ausführungsform, wobei die Datenformate der empfangenen und übertragenen Daten gezeigt sind, bei denen die internen Vorsatzelemente und Stopp-elemente eliminiert sind. Dabei sind zusätzlich zu den internen Vorsatzelementen die internen Stoppelemente mit Ausnahme am Ende des zweiten Verstärkungssignals und des Warnsignals für die Daten eliminiert, die an den Servosystemen 3 oder den Spindelsystemen 4 von Fig. 18 empfangen und davon übertragen werden. Dadurch können die empfangenen und die übertragenen Daten um vier bzw. drei Bytes komprimiert werden. Dann ist die Anzahl n der Spindeln oder Achsen, die innerhalb eines IT

(eines vorbestimmten Steuerzeitintervalls) gesteuert werden können:

$$\begin{aligned} & (2 + 1) \times 10 + 21n \times 10 + (2 + 1) \times 10 + 19n \times 10 \\ 5 \quad & = 2\,000\,000 \times 0,002 \end{aligned} \quad (4).$$

Durch Vereinfachung der obigen Gleichung erhält man:

$$10 \quad 60 + 400n = 4000 \quad (4')$$

und somit ungefähr

$$n = 9,85 \quad (4'')$$

15 Somit können bis zu neun Achsen gesteuert werden.

Fig. 14 ist ein Diagramm einer weiteren Ausführungsform und zeigt die Datenformate der empfangenen und übertragenen Daten ähnlich Fig. 13, wobei jedoch die Datenwörter weiter komprimiert sind. Manchmal kommt es vor, daß die Werte für die Geschwindigkeitsdaten, das Verstärkungssignal, die Geschwindigkeitsrückführdaten usw. nicht, wie in Fig. 9 angegeben, zwischen 0 und dem Maximalwert liegen, und daher kann die Anzahl Bytes, die zur Darstellung der jeweiligen Datenwörter benötigt wird, verringert werden. Außerdem können die Datenwörter gekürzt werden. Beispielsweise für solche Datenwörter, für die die Feinauflösung nicht erforderlich ist (beispielsweise für die Geschwindigkeitsdaten, die Geschwindigkeitsrückführdaten, die Lastrückführdaten usw.) können die Datenwerte beispielsweise auf das nächstliegende Vielfache von 2 gerundet werden: 0, 2, 4, 6, 8, ..., 2n, wobei die Zahlen jeweils um zwei abgestuft sind. Indem man dann diese Zahlen durch zwei dividiert, können die maximalen Datenwerte um die Hälfte verringert werden. Für diejenigen Datenwörter, wie etwa das Verstärkungssignal, bei denen der Bereich der angewandten Werte in Abhängigkeit von der betroffenen Maschine begrenzt ist, kann die Anzahl von einzelnen Datenrahmen praktisch auf beispielsweise

1000 begrenzt werden, ohne daß sich nachteilige Auswirkungen ergeben. Dann kann die Anzahl von Bytes, die zur Darstellung von solchen Datenwörtern benötigt werden, noch weiter als bei den in Fig. 9 gezeigten Werten verringert werden. Durch die obigen Kürzungen oder die Komprimierung von Datenwörtern können die an den Servosystemen 3 und den Spindelsystemen 4 der Fig. 18 empfangenen und übertragenen Daten weiter um zwei bzw. vier Bytes komprimiert werden.

Dann ist die Anzahl n von Spindeln oder Achsen, die innerhalb eines IT (eines vorbestimmten Steuerzeitintervalls) gesteuert werden kann:

$$(2 + 1) \times 10 + 19n \times 10 + (2 + 1) \times 10 + 17n \times 10 = 2\,000\,000 \times 0,002 \quad (5).$$

Durch Vereinfachung der obigen Gleichung erhält man:

$$60 + 360n = 4\,000 \quad (5')$$

und somit ungefähr

$$n = 10,94 \quad (5'').$$

Somit können bis zu zehn Achsen gesteuert werden.

Fig. 15 ist ein Diagramm, das das Format der Daten zeigt, die innerhalb eines vorbestimmten Steuerzeitintervalls unter Anwendung der Datenformate der Ausführungsform von Fig. 14 empfangen und übertragen werden können.

Als nächstes werden unter Bezugnahme auf die Fig. 1 bis 4 die Abläufe der Datenübertragung und Datenkomprimierung beschrieben. Fig. 1 ist ein Flußdiagramm, das den Ablauf für die Datenübertragung zeigt, wobei wiederum Fig. 18 als Beispiel eines automatisierten Fabriksystems dient. Fig. 2 ist ein Flußdiagramm, das den Ablauf für den Datenempfang an den Servosy-

stem 3 und den Spindelsystemen 4 von Fig. 18 zeigt. Fig. 3 ist ein Flußdiagramm, das den Ablauf für die Datenübertragung von den Servosystemen 3 und den Spindelsystemen 4 zu der Hauptsteuereinheit 1 von Fig. 18 zeigt. Fig. 4 ist ein Flußdiagramm, das den Ablauf für den Datenempfang an der Hauptsteuereinheit 1 von Fig. 18 zeigt.

In Fig. 1 wird der Ablauf an der Hauptsteuereinheit 1 in Schritt S1 gestartet. In Schritt S2 wird abgefragt, ob der IT-Startzeitpunkt vorliegt. Wenn die Antwort in Schritt S2 positiv ist, geht der Ablauf zu Schritt S3 weiter. Ansonsten sitzt die Ausführung in einer Warteschleife, bis der IT-Startzeitpunkt vorliegt. In Schritt S3 werden die Daten und Parameter aus dem Speicher 102 ausgelesen, in dem die übertragenen Daten und Parameter gespeichert sind. In Schritt S4 wird die Anzahl von gesteuerten Achsen n auf Eins ($n = 1$) initialisiert, und der Ablauf geht zu Schritt S5 weiter.

In Schritt S5 wird abgefragt, ob Daten für das aktuelle IT verbleiben. Wenn Daten verbleiben, geht der Ablauf zu Schritt S6 weiter. (Wenn nach mehreren Ausführungszyklen der folgenden Schritte S6 bis S14 die Antwort in Schritt S5 negativ wird, geht der Ablauf zu Schritt S15 weiter, wie noch beschrieben wird.) In Schritt S6 wird abgefragt, ob Daten für die n -te Achse verbleiben. Wenn die Antwort in Schritt S6 negativ ist (wenn also keine Daten für die n -te Achse verbleiben), springt die Ausführung zu Schritt S5 zurück nach Inkrementierung der Anzahl n in Schritt S7. Wenn andererseits die Antwort in Schritt S6 positiv ist, geht der Ablauf zu Schritt S8 weiter.

In Schritt S8 wird abgefragt, ob eine Datentabelle zur Konvertierung des aktuellen Datenelements verfügbar ist. Wenn eine Datentabelle existiert, wird das Datenwort durch den Datentabellennachschlagvorgang verarbeitet; ansonsten wird es durch die Softwarelogik, die mit einem Programm implementiert ist, verarbeitet. In dem ersten Ausführungszyklus wird abgefragt, ob eine Datentabelle für das erste Datenelement (die Positi-

onsdaten im Fall dieser Ausführungsform) für die erste Achse existiert. Wenn die Datentabelle existiert, geht die Ausführung zu Schritt S9 weiter. Andernfalls geht die Ausführung zu Schritt S10 weiter.

5

In Schritt S10 werden die Dezimalzahlen (auf der Basis von 10) zuerst in dezimalcodierte Zahlen auf der Basis von 36 beispielsweise mit Hilfe des wohlbekannten Divisionsalgorithmus konvertiert, der mit einem Programm implementiert ist. Dabei werden die Zahlen wiederholt mittels eines Programms durch 36 dividiert, bis der Dividend kleiner als 36 wird, und die aufeinanderfolgenden Reste werden erhalten. Diese Reste, die in umgekehrter Reihenfolge angeordnet sind (so daß der erste erhaltene Rest, der die niederwertigste Ziffer darstellt, am Ende erscheint), repräsentieren die dezimalcodierten Zahlen auf der Basis von 36. Die dezimalcodierten Zahlen auf der Basis von 36, die so erhalten sind, werden dann in Zahlen auf der Basis von 36 konvertiert unter Anwendung der Korrelation der dezimalcodierten Ziffern auf der Basis von 36 auf die Codes auf der Basis von 36, wie in Fig. 10 gezeigt ist.

10

15

20

In Schritt S9 wird abgefragt, ob eine schrittweise Konvertierungstabelle für die Datenkomprimierung existiert. Wenn die Antwort positiv ist, geht der Ablauf zu Schritt S12 weiter. Andernfalls geht der Ablauf zu Schritt S11 weiter.

25

In Schritt S11 werden die in Form von Basis 10 dargestellten Datenwerte in entsprechende Codes auf der Basis von 36 mittels einer Konvertierungsnachschlagtabelle konvertiert. Fig. 5 ist eine Tabelle, die die Konvertierung der in Form von Basis 10 dargestellten Zahlen in entsprechende Codes auf der Basis von 36 zeigt, wobei die ganz rechte Spalte die sieben höchstwertigen Ziffern der Zahlen auf der Basis von 10 und die Kopfzeile die niederwertigste Ziffer davon zeigt. Die Zahlen in dem Code auf Basis 36 sind in jeweiligen Einträgen gezeigt. So können die Zahlen, die auf Basis 10 dargestellt sind, in entspre-

30

35

chende Codes auf der Basis von 36 konvertiert werden mittels der Konvertierungstabelle von Fig. 5. Beispielsweise kann der Code 47 (Basis 36) auf der Basis von 36, der der Zahl 151 (Basis 10) entspricht, aus dem Eintrag am Schnittpunkt der Zeile an der höchstwertigen Ziffer 0000015 (Basis 10) und der Spalte an der niederwertigsten Ziffer 1 (Basis 10) wiedergewonnen werden.

Aufgrund der Hardware-Beschränkungen für die Speicherorganisation der Hauptsteuereinheit 1 von Fig. 18 kann es schwierig sein, die Datentabelle in einer Tabellenform gemäß Fig. 5 zu implementieren. Dann kann die Datentabelle in einem Speicher in Form einer linearen Anordnung wie folgt gespeichert werden. Dabei wird angenommen, daß die oberste Adresse der Datentabelle 10000 ist, so daß die jeweiligen Einträge der Datentabelle an Adressen in aufsteigender Folge gespeichert werden. Beispielsweise wird der Basis-36-Code 47 (Basis 36), der der Zahl 151 (Basis 10) entspricht, an der folgenden Adresse gespeichert:

$10\ 000 + 151 = 10\ 151$ (Basis 10).

Durch Auslesen des bei 10151 gespeicherten Datenworts kann dann der Basis-36-Code 47 (Basis 36) wiedergewonnen werden. Gleichermäßen kann jeder Code auf der Basis von 36, der einer Dezimalzahl entspricht, durch Auslesen der Adresse zurückgewonnen werden, die durch Addition der Dezimalzahl zu der Startadresse der Datentabelle erhalten wird.

Dagegen werden in Schritt S12 die Werte der Daten, die in Codes auf der Basis von 10 dargestellt sind, in entsprechende Codes auf der Basis von 36 mittels einer schrittweisen Datentabelle konvertiert, um die Daten zu komprimieren. Fig. 7 zeigt eine schrittweise Konvertierungstabelle zur Konvertierung der sechsstelligen Zahlen auf Basis 10 in Codes auf der Basis 36 in Schritten von zwei, wobei die ganz rechte Spalte

die fünf höchstwertigen Stellen der Zahlen auf der Basis von 10 und die Kopfzeile die niederwertigste Stelle davon zeigt. Die jeweiligen Einträge repräsentieren die Codes auf der Basis 36, die den Zahlen auf der Basis 10 entsprechen. Diese Codes
5 auf der Basis 36 werden wie folgt erhalten. Zuerst werden die Zahlen auf der Basis von 10 auf eine gerade Zahl gerundet, d.h. die Zahlen auf der Basis von 10 werden durch zwei dividiert, und die resultierenden Quotienten werden gerundet. Als
10 nächstes wird jeder gerundete Quotient in Basis 10 nacheinander durch 36 dividiert, so daß aufeinanderfolgende Reste gebildet werden. Diese Reste, die in umgekehrter Reihenfolge angeordnet sind, repräsentieren die dezimalcodierte Zahl auf der Basis von 36, die dem gerundeten Quotienten entspricht. Somit wird die dezimalcodierte Zahl auf der Basis von 36 weiter in
15 den entsprechenden Code auf der Basis 36 konvertiert. In Schritt S12 werden die Zahlen auf der Basis von 10 in die entsprechenden Codes auf der Basis von 36 durch den Tabellennachschlagvorgang konvertiert, wobei die Tabelle von Fig. 7 genutzt wird. Somit wird beispielsweise die Zahl 21 (Basis 10)
20 zu B (Basis 36) konvertiert.

Im Fall der schrittweisen Datentabelle von Fig. 7 sind die Zahlen um zwei abgestuft. Wenn eine höhere Datenkomprimierung gewünscht wird, können um drei oder mehr abgestufte Zahlen angewandt werden. Wenn andererseits die Charakteristik des Werts
25 eines Datenworts nichtlinear wird oder eine höhere Auflösung in einem bestimmten Bereich des Datenwerts gewünscht wird, kann die Übereinstimmung von den Zahlen auf Basis 10 mit den Codes auf Basis 36 in einem solchen Bereich modifiziert werden, um beispielsweise die Eins-zu-Eins-Übereinstimmung darin
30 zu sichern.

Nach den Schritten S10, S11 und S12 geht der Ablauf zu Schritt S13, in dem die Übertragungsdaten formatiert werden. Dabei
35 werden das Vorsatzelement und das Stoppelement zu dem Kopf und dem Ende des Datenworts hinzugefügt. Wie oben unter Bezugnahme

auf die Fig. 12 und 13 beschrieben wird, können die internen Vorsatzelemente und die Stoppelemente abgekürzt werden. Nach Schritt S13 geht der Ablauf zu Schritt S14 weiter, in dem die formatierten Datenwörter für die aktuelle Achse n in dem Puffer innerhalb des Speichers 102 gespeichert werden. Danach springt der Ablauf zu Schritt S6 zurück, um die Schritte S6 bis S14 zu wiederholen, bis sämtliche Datenwörter für die aktuelle Achse n verarbeitet und in dem Speicher 102 gespeichert sind.

Wenn in Schritt S6 bestimmt wird, daß die Daten für die aktuelle Achse n (die erste Achse im Fall des ersten Ausführungszyklus der Schritte S6 bis S14) sind, wird die Zahl n, die die aktuelle Achse repräsentiert, in Schritt S7 inkrementiert, und die Ausführung springt zu Schritt S5 zurück, in dem abgefragt wird, ob Datenwörter für das aktuelle IT verbleiben. Wenn in Schritt S5 die Antwort positiv ist, wird der Ausführungszyklus der Schritte S6 bis S14 für die Achse n wiederholt.

Wenn in Schritt S5 schließlich bestimmt wird, daß keine Datenwörter für das aktuelle IT verbleiben, geht der Ablauf zu Schritt S15 weiter, in dem die Daten, die formatiert und in dem Puffer innerhalb des Speichers 102 gespeichert sind, zu den Servosystemen 3 oder den Spindelsystemen 4 synchron mit den Übertragungszeitpunkten übertragen werden. Danach wird n rückgesetzt ($n = 0$), und die Ausführung springt zu Schritt S2 zurück, in dem auf den Startzeitpunkt des nächsten IT gewartet wird. So wird der Ablauf der Schritte S2 bis S15 in der Hauptsteuereinheit 1 für jedes IT-Intervall wiederholt.

Das Flußdiagramm von Fig. 2 zeigt den Ablauf für den Datenempfang an den Servosystemen 3 und den Spindelsystemen 4 von Fig. 18. In Schritt S16 werden die Daten und Parameter für die jeweiligen Achsen, die von der Hauptsteuereinheit 1 über das Lichtleiterkabel 6 übertragen werden, in dem Puffer innerhalb des Speichers 302 der Servosysteme 3 oder des Speichers 402

des Speichers 302 der Servosysteme 3 oder des Speichers 402 der Spindelsysteme 4 gespeichert. In Schritt S17 lesen die Servosysteme 3 oder die Spindelsysteme 4 die Daten nach dem Vorsatzelement für die jeweiligen Achsen aus und fragen ab, ob
 5 unverarbeitete Daten in dem Puffer verbleiben. Wenn Daten verbleiben, geht der Ablauf zum nächsten Schritt S18. Andernfalls geht der Ablauf zu Schritt S26, in dem auf den Startzeitpunkt des nächsten IT gewartet wird.

10 In Schritt S18 werden die Daten und Parameter, die nach dem Vorsatzelement liegen, aus dem Puffer innerhalb des Speichers 302 oder 402 ausgelesen. In Schritt S19 wird abgefragt, ob eine Datentabelle verfügbar ist, um das aktuelle Datenelement zu konvertieren. Wenn eine Datentabelle existiert, wird das
 15 Datenwort durch den Datentabellennachschlagvorgang verarbeitet; andernfalls wird es von der Softwarelogik, die mit einem Programm implementiert ist, verarbeitet. Wenn die Datentabelle existiert, geht der Ablauf zu Schritt S20. Andernfalls geht der Ablauf zu Schritt S21.

20 In Schritt S21 werden die Zahlencodes auf der Basis 36 in entsprechende Dezimalzahlen (auf Basis 10) mittels eines Programms zurückgebracht. Das ist der umgekehrte Vorgang der Konvertierung in Schritt S10 in Fig. 1. So kann beispielsweise
 25 der Maximalwert der Positionsdaten: 1NJCHR (Basis 36) in eine entsprechende Dezimalzahl wie folgt zurückgebracht werden: Zuerst wird jede Ziffer (1 bis Z) des Codes auf der Basis von 36 durch entsprechende dezimalcodierte Zahlen auf der Basis von 10 dargestellt. Fig. 11 ist eine Konvertierungstabelle von
 30 Ziffern des Codes auf Basis 36 in dezimalcodierte Zahlen auf Basis 36. So wird der obige Code 1NJCHR in Basis 36 zu der dezimalcodierten Zahl auf Basis 36 konvertiert: [1] [23] [19] [12] [17] [27]. Weiterhin wird die entsprechende Dezimalzahl aus der dezimalcodierten Darstellung in Basis 36 wie folgt erhalten:
 35

$$\begin{aligned}
 & 1 \times 36^5 + 23 \times 36^4 + 19 \times 36^3 + 12 \times 36^2 + 17 \times 36^1 + 27 \times 36^0 \\
 & = 60\,466\,176 + 38\,631\,168 + 886\,464 + 11\,552 + 612 + 27 \\
 & = 99\,999\,999 \qquad \qquad \qquad (\text{Basis } 10).
 \end{aligned}$$

5 In Schritt S20 wird abgefragt, ob eine schrittweise Datentabelle für die Datenkomprimierung existiert. Wenn die Antwort positiv ist, geht der Ablauf zu Schritt S23 weiter. Andernfalls geht der Ablauf zu Schritt S22 weiter. In Schritt S22 werden die Datenwerte, die in dem Code auf Basis 36 dargestellt sind, in entsprechende Zahlen auf Basis 10 durch Nachschlagen in einer Konvertierungstabelle konvertiert. Fig. 6 ist eine Tabelle, die die Konvertierung der Zahlen, die in Basis 36 dargestellt sind, in entsprechende Dezimalzahlen zeigt, wobei die ganz rechte Spalte die fünf höchstwertigen Stellen der Codezahlen auf der Basis von 36 und die Kopfzeile die niederwertigste Stelle davon zeigt. Die entsprechenden Dezimalzahlen sind in jeweiligen Einträgen gezeigt. Somit können die in Basis 36 dargestellten Zahlen in entsprechende Dezimalzahlen mittels der Konvertierungstabelle von Fig. 6 konvertiert werden. Beispielsweise kann die Dezimalzahl 151, die dem Code 000047 (Basis 36) entspricht, aus dem Eintrag der Konvertierungstabelle am Schnittpunkt der Zeile an den fünf höchstwertigen Stellen 00004 (Basis 36) und der Spalte an der niederwertigsten Stelle 7 (Basis 36) rückgewonnen werden. Die Konvertierungstabelle von Fig. 6 kann wie im Fall der Tabelle von Fig. 5 als eine lineare Anordnung implementiert werden.

In Schritt S23 dagegen werden die Werte der in Basis 36 dargestellten Daten, die auf der Basis der schrittweisen Konvertierung von Zahlen auf Basis 10 in Schritt S12 in Fig. 1 erhalten werden, zu entsprechenden Codes auf Basis 10 mittels einer rückschreitenden Datentabelle rückgewonnen. Fig. 8 zeigt eine rückschreitende Konvertierungstabelle zur Rückführung der schrittweisen Codes auf Basis 36, wie sie unter Anwendung der Tabelle von Fig. 7 erhalten sind, in entsprechende sechsstellige Dezimalzahlen in zwei Schritten, wobei die ganz rechte

Spalte die zwei höchstwertigen Stellen der Codezahlen auf Basis 36 und die Kopfzeile die niederwertigsten Stellen davon zeigt. Die jeweiligen Einträge der Tabelle repräsentieren die Dezimalzahlen, die den Codes auf der Basis 36 entsprechen. Somit wird beispielsweise der Basis-36-Code B (Basis 36) in die entsprechende Dezimalzahl 22 (Basis 10) durch Nachschlagen in der Tabelle an dem Eintrag am Schnittpunkt der Zeile bei der höchstwertigen Stelle 00 und der Spalte bei der niederwertigsten Stelle B konvertiert. Es ist zu beachten, daß die schrittweisen Konvertierungstabellen (z.B. die Tabelle von Fig. 7) innerhalb der Hauptsteuereinheit 1 zur Konvertierung der Zahlen auf der Basis 10 in Codes auf der Basis 36 und die umgekehrt schrittweisen Konvertierungstabellen (z.B. die Tabelle von Fig. 8) innerhalb der Servosysteme 3 oder der Spindelsysteme 4 zur Rückgewinnung der Codes auf der Basis 36 in die Zahlen auf der Basis 10 gepaart sind, so daß die Datenwerte, die zuerst mit Hilfe von schrittweisen Konvertierungstabellen zu Codes auf der Basis 36 komprimiert und dann in den komprimierten Codes auf Basis 36 von der Steuereinheit 1 übertragen werden, in die ursprünglichen Dezimalwerte durch die Konvertierung durch Nachschlagen in den Umkehrtabellen an den Servosystemen 3 oder den Spindelsystemen 4 rückgewonnen werden.

Nach den Schritten S21, S22 und S23 geht der Ablauf zu Schritt S24, in dem die empfangenen Daten formatiert werden. Wenn die Vorsatz- und/oder Endelemente auf der Seite der Hauptsteuereinheit 1 vor der Übertragung gekürzt werden, werden die erforderlichen Vorsatz- und/oder Endelemente entsprechend den Werten der zugehörigen Parameter hinzugefügt, um die vorbestimmten Datenformate zu erhalten. Dabei kann die Formatierung der empfangenen Daten unmittelbar nach Schritt S16 und nicht erst nach den Datenkonvertierungsschritten S21 bis S23 durchgeführt werden.

Als nächstes werden in Schritt S25 die formatierten Datenwörter in dem Puffer innerhalb des Speichers 302 oder 402 gespeichert. Danach springt der Ablauf zu Schritt S17 zurück, um die Schritte S18 bis S25 zu wiederholen, bis sämtliche Datenwörter für das aktuelle IT verarbeitet und in dem Pufferspeicher gespeichert sind. Wenn die Antwort in Schritt S17 schließlich negativ wird (d.h. wenn sämtliche empfangenen Daten für das aktuelle IT verarbeitet sind), geht der Ablauf zu Schritt S26, in dem auf den Startzeitpunkt des nächsten IT gewartet wird. Wenn der Startzeitpunkt des nächsten IT eintritt, springt der Ablauf zu Schritt S16 zurück, um die Schritte S17 bis S26 zu wiederholen. Wenn die jeweiligen Servosysteme 3 oder Spindelsysteme 4 die empfangenen Daten benötigen, werden die notwendigen Daten aus dem Puffer in dem Speicher 302 oder 402 ausgelesen, in dem die Daten in Schritt S25 gespeichert wurden.

Jedesmal, wenn die Daten an den jeweiligen Spindeln oder Achsen abgetastet werden und die daran ausgeführten Berechnungen innerhalb der Servosysteme 3 oder der Spindelsysteme 4 beendet sind, werden die erhaltenen Daten in dem Speicher 302 oder 402 gespeichert. Die Daten werden von dort zu der Hauptsteuereinheit 1 innerhalb der jeweiligen ITs übertragen. Als nächstes wird das Verfahren der Datenkonvertierung und -übertragung von den Servosystemen 3 und den Spindelsystemen 4 zu der Hauptsteuereinheit 1 unter Bezugnahme auf die Fig. 3 und 4 beschrieben.

Fig. 3 ist ein Flußdiagramm, das den Ablauf der Datenübertragung von den Servosystemen 3 und den Spindelsystemen 4 zu der Hauptsteuereinheit 1 von Fig. 18 zeigt. In Fig. 3 wird der Ablauf an den Servosystemen 3 oder den Spindelsystemen 4 in Schritt S30 gestartet. In Schritt S31 wird abgefragt, ob ein IT-Startzeitpunkt vorliegt. Wenn die Antwort in Schritt S31 positiv ist, geht der Ablauf zu Schritt S32 weiter. Andernfalls sitzt der Ablauf in einer Warteschleife, bis der IT-

17.10.97

Startzeitpunkt ankommt. In Schritt S32 werden die Daten und Parameter aus dem Speicher 302 oder 402 ausgelesen, in dem die übertragenen Daten und Parameter abgelegt sind.

5 In Schritt S33 wird abgefragt, ob Daten für das aktuelle IT verbleiben. Wenn Daten verbleiben, geht der Ablauf zu Schritt S34, in dem abgefragt wird, ob eine Datentabelle verfügbar ist, um das aktuelle Datenelement zu konvertieren. Wenn eine Konvertierungsdatentabelle vorhanden ist, wird das Datenwort
10 durch den Datentabellennachschlagprozeß verarbeitet; andernfalls wird es von der Softwarelogik, die durch ein Programm implementiert ist, verarbeitet.

Wenn die Antwort in Schritt S34 positiv ist, geht der Ablauf
15 zu Schritt S36, in dem die Dezimalzahlen (Basis 10) in entsprechende Zahlen auf Basis 36 durch ein Programm, das die Softwarelogik implementiert, konvertiert werden. Das Konvertierungsverfahren gleicht demjenigen in Schritt S10 in Fig. 1.

20 In Schritt S35 wird abgefragt, ob eine schrittweise Konvertierungsdatentabelle für die Datenkomprimierung vorhanden ist. Wenn die Antwort positiv ist, geht der Ablauf zu Schritt S38. Andernfalls geht der Ablauf zu Schritt S37. In Schritt S37 werden die in Basis 10 dargestellten Datenwerte in entsprechende Codes auf Basis 36 mittels einer Konvertierungsnachschlagtabelle konvertiert. Diese Konvertierung gleicht derjenigen in Schritt S11 von Fig. 1, die oben beschrieben wurde.
25 In Schritt S38 dagegen werden die Werte der Daten, die in Basis 10 dargestellt sind, in entsprechende Codes auf Basis 36
30 mittels einer schrittweisen Datentabelle für die Komprimierung der Daten konvertiert. Die schrittweise Konvertierung der Zahlen auf Basis 10 in entsprechende komprimierte Codes auf Basis 36 gleicht derjenigen in Schritt S12 von Fig. 1.

Nach den Konvertierungsschritten S36, S37 und S38 geht der Ablauf zu Schritt S39, in dem die Übertragungsdaten ähnlich wie in Schritt S13 von Fig. 1 formatiert werden.

5 In Schritt S40 werden dann die formatierten Datenwörter in dem Puffer innerhalb des Speichers 302 oder 402 gespeichert. Danach springt der Ablauf zu Schritt S33 zurück, um die Schritte S33 bis S40 zu wiederholen, bis sämtliche Datenwörter für das aktuelle IT verarbeitet und in dem Speicher 302 oder 402 abgelegt sind.

10 Wenn in Schritt S33 festgestellt wird, daß für das aktuelle IT keine Datenwörter verbleiben, geht der Ablauf zu Schritt S41, in dem die formatierten und in dem Puffer innerhalb des Speichers 302 oder 402 gespeicherten Daten zu der Hauptsteuereinheit 1 synchron mit den Übertragungszeitpunkten übertragen werden. Danach springt der Ablauf zu Schritt S31 zurück, in dem auf den Startzeitpunkt des nächsten IT gewartet wird. So wird der Ablauf der Schritte S31 bis S41 für jedes IT-Intervall wiederholt.

15 Fig. 4 ist ein Flußdiagramm, das den Ablauf für den Datenempfang an der Hauptsteuereinheit 1 von Fig. 18 zeigt. In Schritt S42 werden die Gruppen von Daten, die über die Servosysteme 3 übertragen werden, in dem Puffer innerhalb des Speichers 102 der Hauptsteuereinheit 1 entsprechend den jeweiligen Achsen gespeichert.

25 In Schritt S43 wird zuerst die aktuelle Zahl n von Achsen auf Eins initialisiert ($n = 1$). In Schritt S44 wird abgefragt, ob für das aktuelle IT Daten verbleiben. Wenn Daten verbleiben, geht der Ablauf zu Schritt S45 weiter. In Schritt S45 wird abgefragt, ob Daten für die n -te Achse verbleiben. Wenn in Schritt S45 die Antwort negativ ist (wenn also keine Daten für die n -te Achse verbleiben), springt die Ausführung zu Schritt

S44 zurück nach Inkrementierung der Zahl n in Schritt S46. Wenn dagegen in Schritt S45 die Antwort positiv ist, geht der Ablauf zu Schritt S47 weiter.

- 5 In Schritt S47 wird abgefragt, ob eine Datentabelle verfügbar ist, um das aktuelle Datenelement zu konvertieren. Wenn eine Datentabelle existiert, wird das Datenwort durch den Datentabellennachschlagprozeß verarbeitet; andernfalls wird es durch die Softwarelogik verarbeitet, die durch ein Programm implementiert ist. Wenn die Datentabelle vorhanden ist, geht der
10 Ablauf zu Schritt S48. Andernfalls geht der Ablauf zu Schritt S49. In Schritt S49 werden die Dezimalzahlen (Basis 10) zu Zahlen auf Basis 36 mittels eines Programms konvertiert. Das Konvertierungsverfahren gleicht demjenigen in Schritt S21 in
15 Fig. 2. In Schritt S48 wird abgefragt, ob eine schrittweise Datentabelle für die Datenkomprimierung vorhanden ist. Wenn die Antwort positiv ist, geht der Ablauf zu Schritt S51. Andernfalls geht der Ablauf zu Schritt S50. In Schritt S50 werden die in Basis 10 dargestellten Datenwerte in entsprechende
20 Codes auf Basis 36 mittels einer Konvertierungsnachschlagtabelle konvertiert. Dieser Tabellennachschlagschritt gleicht demjenigen in Schritt S22 in Fig. 2. In Schritt 51 dagegen werden die Werte der in Basis 10 dargestellten Daten in entsprechende Codes auf Basis 36 mittels einer schrittweisen Datentabelle für die Datenkomprimierung konvertiert. Diese Kon-
25 vertierung gleicht derjenigen in Schritt S23 in Fig. 2.

- Nach den Schritten S49, S50 und S51 geht der Ablauf zu Schritt S52, in dem die Übertragungsdaten ähnlich wie in Schritt S24
30 formatiert werden. Nach Schritt S52 geht der Ablauf zu Schritt S53, in dem die formatierten Datenwörter für die aktuelle Achse n in dem Puffer innerhalb des Speichers 302 oder 402 abgelegt werden. Danach springt die Ausführung zu Schritt S45 zurück, um die Schritte S47 bis S53 zu wiederholen, bis sämt-

liche Datenwörter für die aktuelle Achse n verarbeitet und in dem Speicher 302 oder 402 abgelegt sind.

5 Wenn in Schritt S45 festgestellt wird, daß für die aktuelle Achse n keine Daten mehr verbleiben, wird in Schritt S46 die Zahl n, die die aktuelle Achse repräsentiert, inkrementiert, und der Ablauf springt zu Schritt S44 zurück, in dem abgefragt wird, ob Datenwörter für das aktuelle IT verbleiben. Wenn in Schritt S44 die Antwort positiv ist, wird der Ausführungszyklus der Schritte S44 bis S53 wiederholt. Wenn schließlich in 10 Schritt S44 festgestellt wird, daß für das aktuelle IT keine Datenwörter verbleiben, geht der Ablauf zu Schritt S54, in dem die formatierten und in dem Puffer innerhalb des Speichers 302 oder 402 gespeicherten Daten zu der Hauptsteuereinheit 1 synchron mit den Übertragungszeitpunkten übertragen werden. Da- 15 nach springt der Ablauf zu Schritt S42 zurück, um den Ablauf wie oben beschrieben zu wiederholen.

Es sollte im übrigen beachtet werden, daß die Abfrage in Schritt S8 und S9 in Fig. 1, in Schritt S19 und Schritt S20 in 20 Fig. 2, in Schritt S34 und Schritt S35 in Fig. 3 sowie in Schritt S47 und S48 in Fig. 4 auf der Basis der Werte der Parameter erfolgen kann, die jeweiligen Datenwörtern zugeordnet sind und mit diesen übertragen werden. Die Werte der Parameter 25 zeigen an, ob die Konvertierung mittels der Softwarelogik, der einfachen Konvertierungstabellen- oder der schrittweisen Konvertierungstabellen-Nachschlagmethode durchzuführen ist. Ferner kann die Feststellung, ob die Vorsatzelemente und die Endelemente für die jeweiligen Datenelemente anzubringen oder 30 zu kürzen sind, auf der Grundlage der Werte von damit zusammenhängenden Parametern erfolgen.

Dann wird beispielsweise die Feststellung in Schritt S13 in Fig. 1, ob das Vorsatzelement und die Endelemente anzubringen 35 oder zu kürzen sind, nach Maßgabe der Werte der Parameter

durchgeführt. Ferner wird beispielsweise die Feststellung in Schritt S24 in Fig. 2, ob das Hinzufügen von Vorsatz- und/oder Endelementen zu den jeweiligen Datenwörtern notwendig ist (d.h. ob die Vorsatz- und/oder die Endelemente auf der Übertragungsseite gekürzt wurden und auf der Empfangsseite hinzugefügt werden sollten), auf der Grundlage der Werte der Parameter durchgeführt, die den jeweiligen Datenwörtern zugeordnet sind.

Gemäß dieser Ausführungsform des Verfahrens unter Anwendung des Systems von Fig. 18 können innerhalb eines IT bis zu zehn Spindeln oder Achsen gesteuert werden. Außerdem verbleibt eine Toleranz von 340 Bits innerhalb jedes IT, auch wenn die maximale Anzahl von Achsen (d.h. zehn Achsen) gesteuert wird. Somit können die Parameter für die Servo- und die Hauptachsen entsprechend dem Bedarf für die Spindeln in dem Toleranzzeitintervall jedes IT übertragen werden.

Im Fall der obigen Ausführungsform werden die Daten über ein einziges Lichtleiterkabel übertragen und empfangen. Die Erfindung ist aber auch bei der Datenübertragung über RS232C anwendbar. Ferner werden im Fall der vorstehenden Ausführungsform die Dezimalwerte in Codes auf Basis 36 komprimiert. Der gleiche Vorteil der Datenkomprimierung wird aber auch erhalten, wenn die Dezimalzahlen in irgendein Zahlensystem konvertiert werden, für das die Basis größer als 10 ist. Entsprechend dem EIA-Standard, nach dem ein Byte aus zehn Bits besteht (10 Bits/Byte), sind die Zahlencodesysteme auf der Basis 128 ($=2^7$) und der Basis 256 ($=2^8$) die Obergrenzen der Basen des Zahlensystems. Wenn jedoch andere Standards angewandt werden, können die Zahlencodesysteme, deren Basis 256 überschreitet, zur Implementierung der Erfindung angewandt werden.

Ferner werden im Fall der obigen Ausführungsform die Daten für jedes IT verarbeitet. Wenn aber die Verarbeitungsgeschwindigkeit der Terminals langsam ist, können die Daten für zwei ITs

17.10.97

5 verarbeitet und von der Hauptsteuereinheit 1 innerhalb eines einzelnen IT übertragen werden. Im Fall der obigen Ausführungsform wird die Datenkonvertierung hauptsächlich mittels Software durchgeführt. Der gleiche Ablauf kann aber teilweise
10 mittels Hardware auf der Basis von Firmware oder vollständig mittels Hardware implementiert werden. Außerdem werden im Fall der obigen Ausführungsform innerhalb der übertragenen Information nur die Datenelementbereiche komprimiert. Die gleiche Komprimierungstechnik kann aber auch bei Vorsatzelementen angewandt werden. Die Vorsatzelemente können ohne weiteres mit-
15 tels eines Datentabellen-Nachschlagverfahrens komprimiert werden.

15 Ferner ist das Verfahren der vorliegenden Erfindung auf die Datenübermittlung innerhalb eines FA- bzw. automatisierten Fabriksystems anwendbar. Fig. 16 ist ein Diagramm, das das System zeigt, das um einen FA-Mastercomputer mit untergeordneten Zellensteuerungen herum zentriert ist, mit denen die verschie-
20 denen Untersysteme gekoppelt sind. Fig. 17 ist ein Diagramm, das das System ähnlich demjenigen von Fig. 16 zeigt, wobei jedoch die Zellensteuerungen weggelassen sind. In Fig. 16 werden beispielsweise die Daten zu und von dem FA-Computer 1000 über die Übertragungsleitungen 1100, 1200-1 bis 1200-3 und die untergeordneten Zellensteuerungen 1020, von und zu den weiteren
25 untergeordneten Vorrichtungsgruppen, wie etwa den numerisch gesteuerten Vorrichtungen 1030, automatischen Lagerhaussteuerungen 1040, PLC-Gruppen 1050, Transporteinrichtungs-Steuerungen 1060, Meßeinrichtungs-Steuerungen 1070 und Fabrikmanagement-Terminals 1080 übertragen. Die Datenübertragung kann auch
30 zwischen den untergeordneten Systemen sowie zu und von dem CAD/CAM-System 1100 stattfinden. Die Datenübertragung in dem System von Fig. 17 gleicht derjenigen des Systems von Fig. 16.

35 Im Fall der Systeme von Fig. 16 und 17 können die Dezimalzahl-
daten, die in einem Zeichendarstellungscodesystem wie etwa dem ASCII-Code dargestellt sind, vor der Übertragung in entspre-

chende Zahlencodes auf höherer Basis konvertiert werden (z.B. die Codes auf der Basis von 36), die in demselben ASCII-Code-system dargestellt sind, und nach dem Empfang in die ursprünglichen Dezimalwerte rückkonvertiert werden, so daß die Daten in komprimierter Form übertragen werden. Das Prinzip der Komprimierung von Daten ist das gleiche wie im Fall der obigen Ausführungsform. Es versteht sich von selbst, daß das Prinzip der Datenkomprimierung auf die Datenübertragung zwischen der Hauptsteuereinheit 1 und dem Displaysystem 2 bei dem System von Fig. 18 anwendbar ist. Weiterhin ist das gleiche Prinzip bei Computersystemen einschließlich solcher auf der Basis von Personalcomputern anwendbar.

Es ist zu beachten, daß die Methode der Konvertierung und Wiederherstellung der Dezimalzahlenwerte in Abhängigkeit von den Systemanforderungen gewählt werden kann. Allgemein erfordert die Implementierung von Softwarelogik weniger Speicherkapazität als die Tabellennachschlagmethode. Andererseits erfordert die Tabellennachschlagmethode weniger Zeit. Der Kompromiß muß bewertet werden, so daß die beiden Methoden gewählt und kombiniert werden, um das Leistungsvermögen und die Kosten gegeneinander abzugleichen.

5

Patentansprüche

1. Verfahren zur seriellen Datenübertragung über Übertra-
gungsleitungen (5, 6) zwischen Informationseinheiten
(1, 2, 3, 4) in einem automatisierten Fabrikssystem, wobei
jedes Datenwort der Daten ein Vorsatzelement, eine Dezi-
malzahl und ein Endelement aufweist und in einem industri-
ellen Standardcode zum Informationsaustausch dargestellt
wird, wobei das Verfahren folgende Schritte aufweist:

- a) Konvertieren (S10, S11, S12) der Dezimalzahlen der
Datenwörter in Zahlencodes auf der Basis n in dem
industriellen Standardcode, wobei die Basis n größer
als 10 ist,
- b) Formatieren (S13) der konvertierten Daten aus dem
Schritt a) durch Bestimmen, welche Vorsatz- und/oder
Endelemente in Abhängigkeit von Parameterwerten der
Datenwörter zu entfernen sind,
wobei die Schritte (a) und (b) in einer Übertragungs-
einheit (1) des automatisierten Fabriksystems durchge-
führt werden,
- c) Übertragen der konvertierten und formatierten Daten-
wörter von der Übertragungseinheit (1) zu einer
Empfangseinheit (2, 3, 4) über die Übertragungsleitun-
gen (5, 6).

2. Verfahren nach Anspruch 1,
das ferner die folgenden Schritte in der Empfangseinheit
aufweist:

17.10.97

- 2 -

d) Zurückkonvertieren (S21, S22, S23) der Zahlencodes auf der Basis n zurück in Dezimalzahlen der Datenwörter und

e) Formatieren (S24) der zurückkonvertierten Daten aus dem Schritt (d) durch erneutes Hinzufügen der Vorsatz- und/oder Endelemente, die beim Schritt (b) in Abhängigkeit von den Parameterwerten der Datenwörter entfernt worden sind.

10 3. Verfahren nach Anspruch 1 oder 2, wobei der industrielle Standardcode zum Informationsaustausch der amerikanische Standardcode für Informationsaustausch ist.

15 4. Verfahren nach einem der Ansprüche 1 bis 3, wobei die Informationseinheiten (1, 2, 3, 4) eine Hauptsteuereinheit (1) und Terminals (2, 3, 4) innerhalb des automatisierten Fabriksystems aufweisen, die durch Übertragungsleitungen (5, 6) verbunden sind, wobei die
20 Terminals (2, 3, 4) numerisch gesteuerte Einheiten aufweisen.

25 5. Verfahren nach Anspruch 4, wobei die Hauptsteuereinheit (1) und die Terminals (2, 3, 4) mit jeweiligen Speichern (102, 202, 302, 402) versehen sind.

30 6. Verfahren nach einem der Ansprüche 1 bis 5, wobei die Informationseinheiten (1, 2, 3, 4) folgendes aufweisen:
einen Hauptcomputer (1000), ein CAD/CAM System (1010), Zellensteuerungen (1020) und Terminaleinheitsgruppen innerhalb eines automatisierten Fabriksystems, die über die Übertragungsleitungen verbunden sind.

35

17.10.97

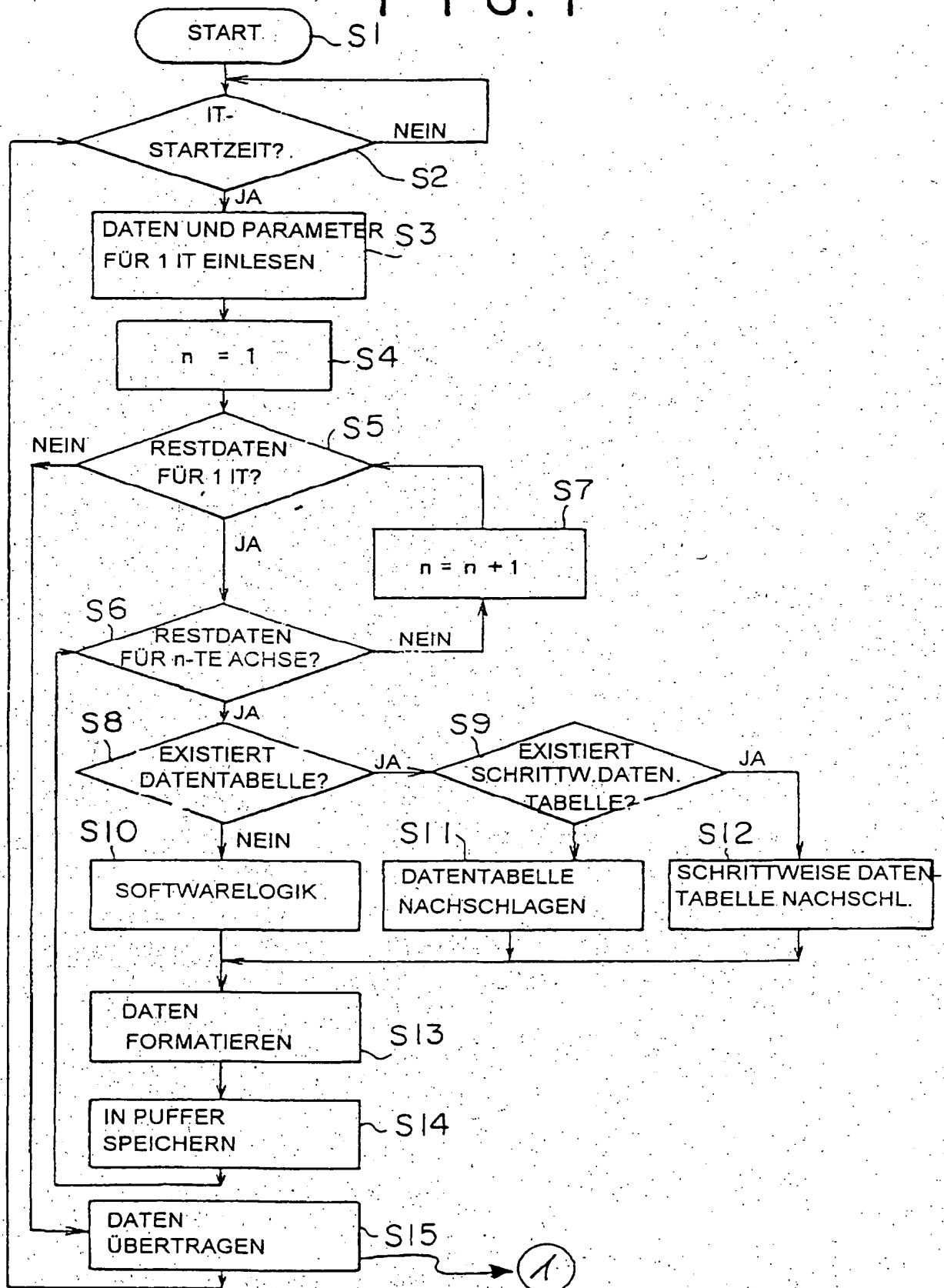
- 3 -

7. Verfahren nach einem der Ansprüche 1 bis 6,
wobei die Konvertierung mittels einer Softwarelogik durch-
geführt wird, die von einem Softwareprogramm implementiert
wird.
8. Verfahren nach einem der Ansprüche 1 bis 6,
wobei die Konvertierung mit einem Tabellennachschlagver-
fahren durchgeführt wird, und zwar auf der Basis einer
Konvertierungstabelle, die eine Korrelation zwischen den
Dezimalzahlen und den Zahlencodes auf der Basis n her-
stellt.
9. Verfahren nach einem der Ansprüche 1 bis 6,
wobei die Konvertierung mit einem Tabellennachschlagver-
fahren mit schrittweiser Konvertierung durchgeführt wird.
10. Verfahren nach einem der Ansprüche 1 bis 9,
wobei das Konvertierungsverfahren für die Konvertierung
von den Dezimalzahlen in Zahlencodes auf der Basis n in
Abhängigkeit von einem Parameter gewählt wird, der jedem
Datenwort zugeordnet ist, wobei das Konvertierungsverfah-
ren aus folgenden Verfahren ausgewählt wird: Software-
logik, die mit einem Programm implementiert wird, Konver-
tierungstabellennachschlagverfahren oder Tabellennach-
schlagverfahren mit schrittweiser Konvertierung.
11. Verfahren nach Anspruch 9 oder 10,
wobei die schrittweise Konvertierungstabelle eine
Korrelation von den Dezimalzahlen, die auf ein nächstes
Vielfaches einer ganzen Zahl gerundet sind, zu den Zahlen-
codes auf der Basis n herstellt, die den gerundeten Dezi-
malzahlen, geteilt durch die ganze Zahl, entsprechen,
wobei die ganze Zahl gleich oder größer als 2 ist.
12. Verfahren nach einem der Ansprüche 1 bis 11,
wobei die jeweiligen Serien von Datenwörtern in jeweiligen
Zeitintervallen (IT) einer vorbestimmten Länge übertragen
werden.

17.10.97

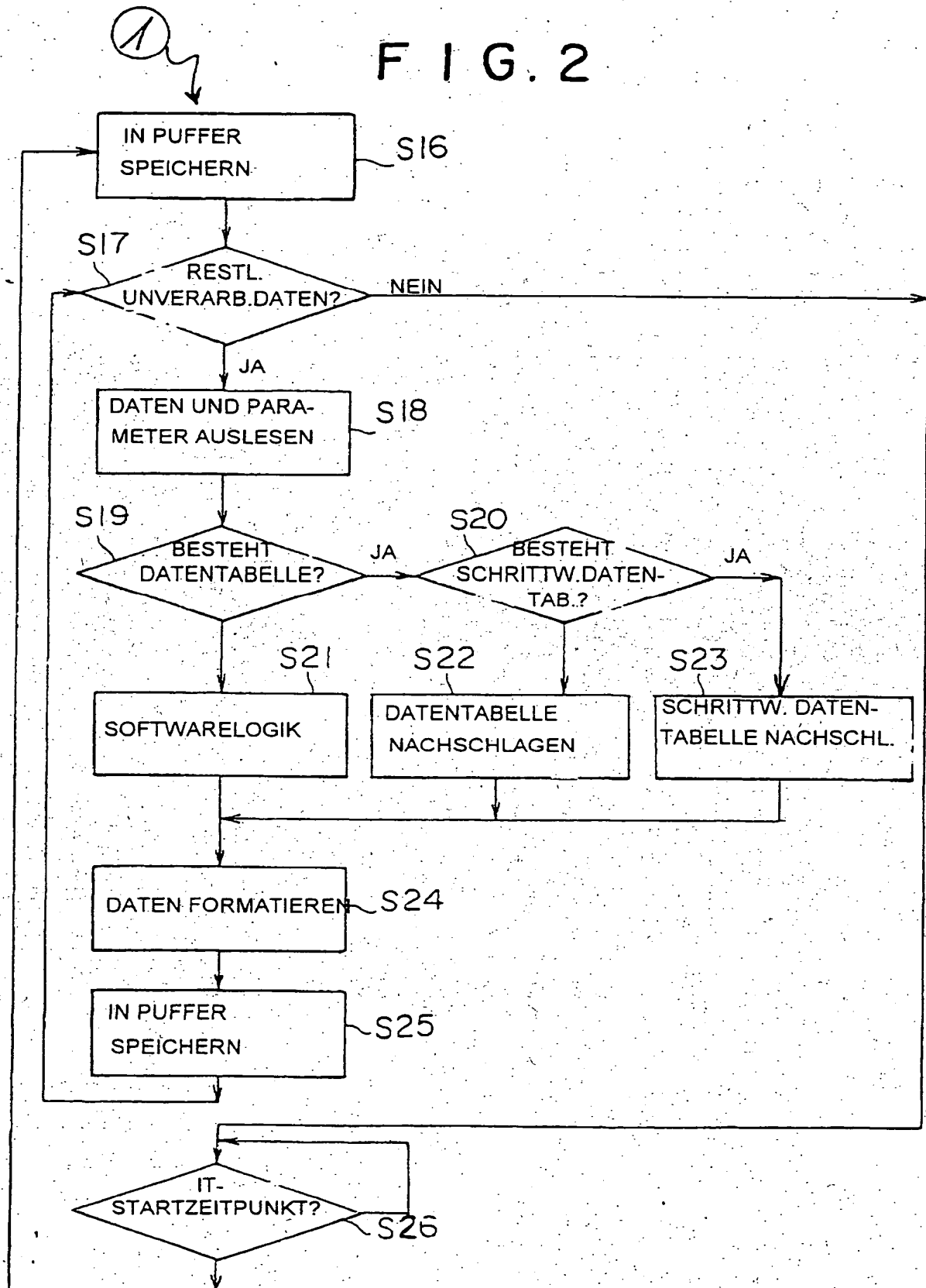
692 21 046.6-08

FIG. 1



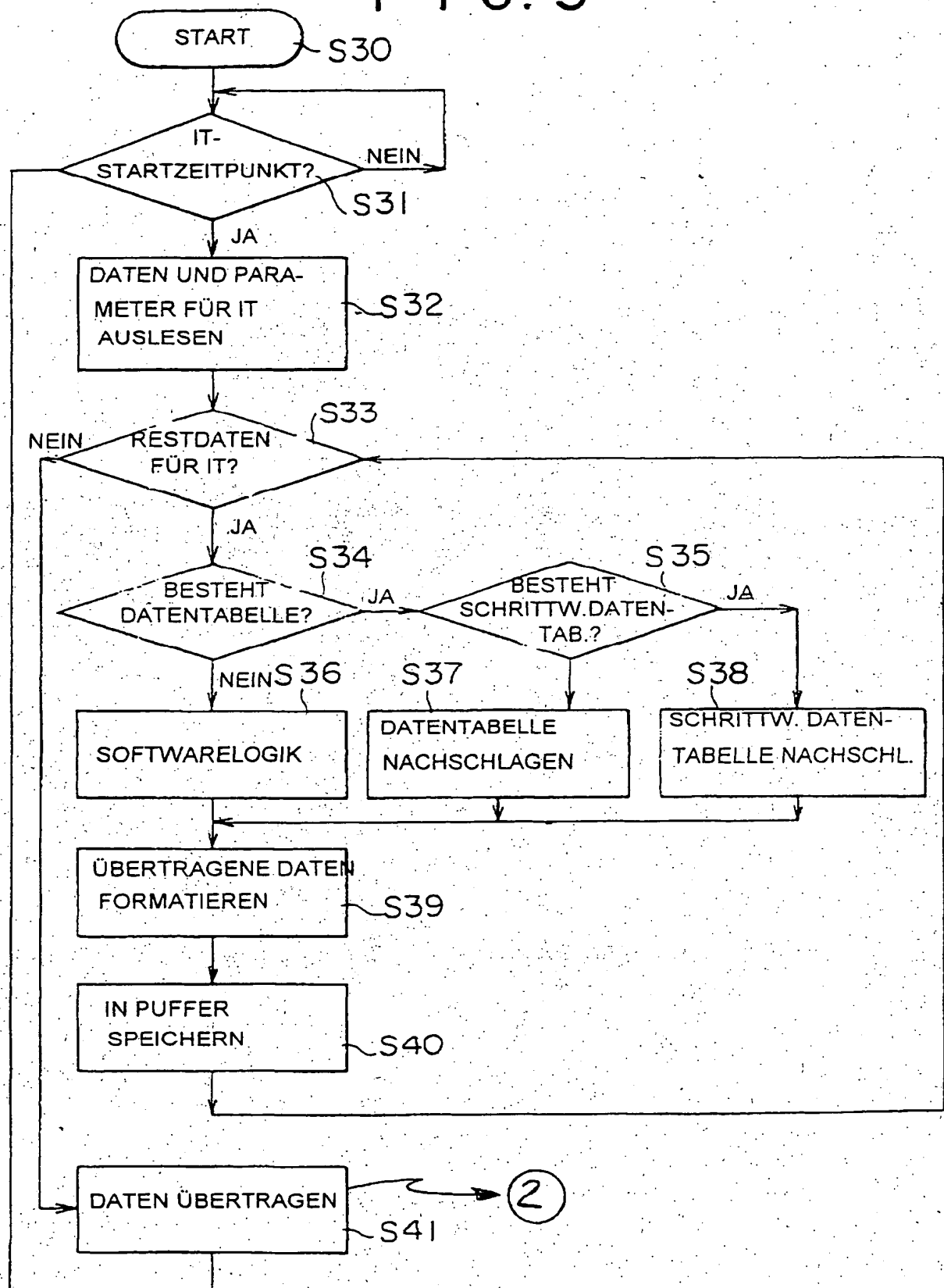
17.10.97

FIG. 2



17.10.97

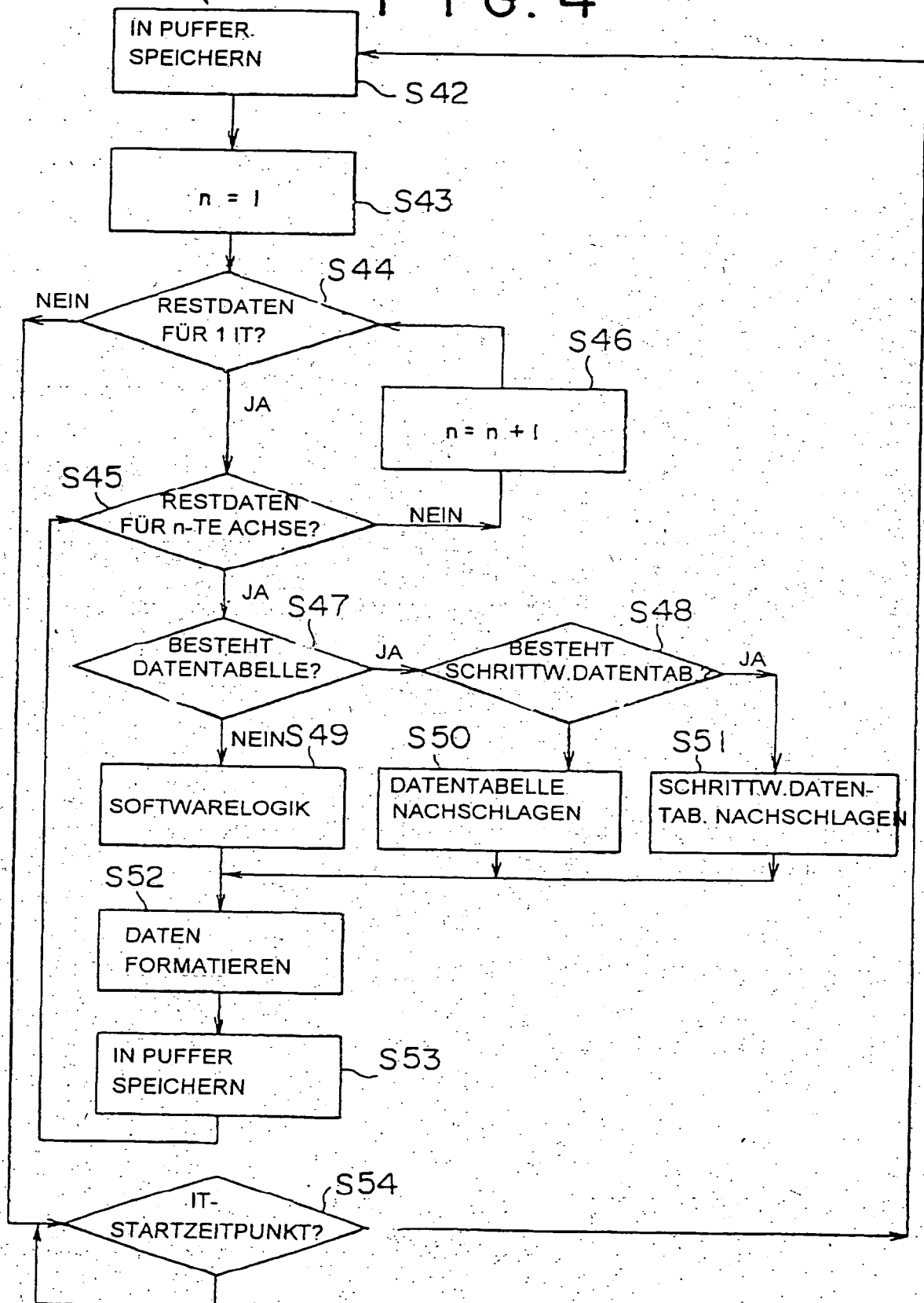
FIG. 3



17.10.97

②

FIG. 4



F I G. 5

[illegible]

FIG. 6

KONVERTIERUNGSTABELLE VON BASIS 36 CODE IN BASIS 10

LSD	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
5MSD:	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
00000	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
00001	36	37	38	39	40	41	42	43	44	45																										
00002																																				
00003																																				
00004																																				
00005																																				
00006																																				
00007																																				

5 10 15 20

FIG. 7

KONVERTIERUNGSTABELLE VON BASIS 10 ZU SCHRITTWEISEM BASIS 36 CODE

LSD 5MSD's	0	1	2	3	4	5	6	7	8	9
00000	0	1	2	3	4	5	6	7	8	9
00001	5	6	7	8	9	A	B	C	D	E
00002	A	B	C	D	E	F	G	H	I	J
00003	F	G	H	I	J	K	L	M	N	O

FIG. 8

KONVERTIERUNGSTABELLE VON SCHRITTWEISEM BASIS 36 CODE ZU BASIS 10

LSD 2MSD's	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	G	H		W	X	Y	Z
00	0	2	4	6	8	10	12	14	16	18	20	22	24	26	28	30	32	34		64	66	68	70
01	72	74	76	78	80	82	84	86												134	136	138	140
02	142	144	146	148	150	152														204	206	208	210
03	212	214	216	218	220	222															276	278	280
																					346	348	350

17.10.97

FIG.9

	BASIS 10		BASIS 36	
	MAXIMUM	BYTES	MAXIMUM	BYTES
(a) POSITIONS DATEN	99999999 (10)	8	1NJCHR(36)	6
(b) GESCHWINDIGKEITS-DATEN	999999 (10)	6	LFLR(36)	4
(c) STEUERDATEN (1)	9 (10)	1	J (36)	1
(d) STEUERDATEN (2)	9 (10)	1		
(e) VERSTÄRKUNGS-SIGNAL (1)	99999 (10)	5	255R(36)	4
(f) VERSTÄRKUNGS-SIGNAL (2)	999 (10)	3	RR(36)	2
GESAMTBYTES VON EMPFANGENEN DATEN		24		17
(g) POSITIONS RÜCK-FÜHRDATEN	99999999 (10)	8	1NJCHR(36)	6
(h) GESCHWINDIGKEITS-RÜCKFÜHRDATEN	999999 (10)	6	LFLR(36)	4
(i) LAST-RÜCKFÜHRDATEN	9999 (10)	4	7PR(36)	3
(j) WARNSIGNAL	999 (10)	3	RR(36)	2
GESAMTBYTES VON ÜBERTRAGENEN DATEN		21		15

N.B. DIE SUFFIXE (10) UND (36) BEZEICHNEN DIE BASIS DER ZAHLEN

FIG.10

KONVERTIERUNGSTABELLE VON DEZIMALCODIERT BASIS 36 IN CODE BASIS 36

DEZIMALCODIERT BASIS 36																																				
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35
CODE BASIS 36	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z

7 5 3 2

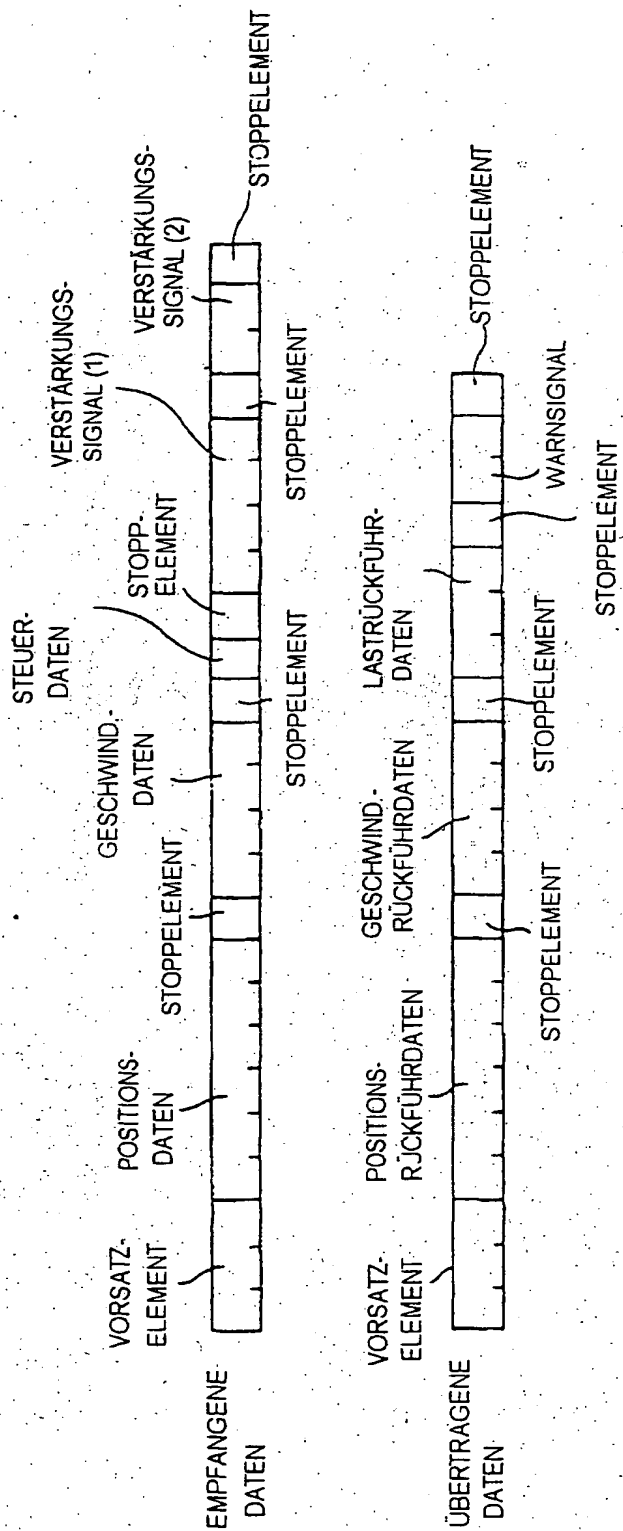
FIG.11

KONVERTIERUNGSTABELLE VON CODE BASIS 36 IN DEZIMALCODIERT BASIS 36

CODE BASIS 36	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
DEZIMALCODIERT BASIS 36	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35

36

FIG. 12



12

FIG. 13

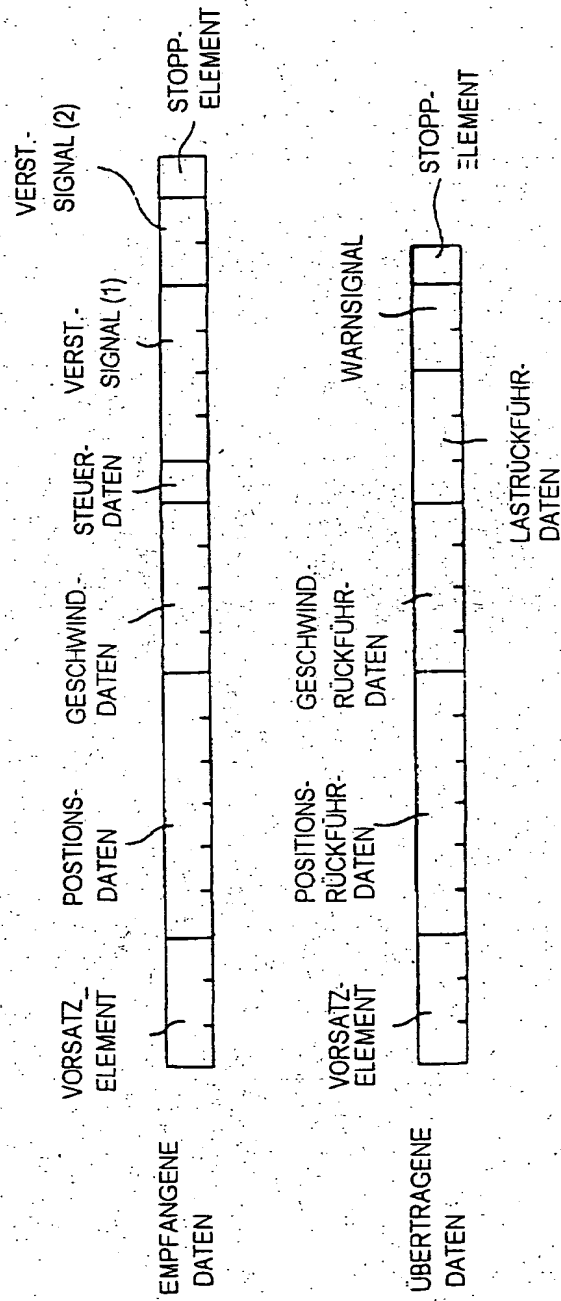
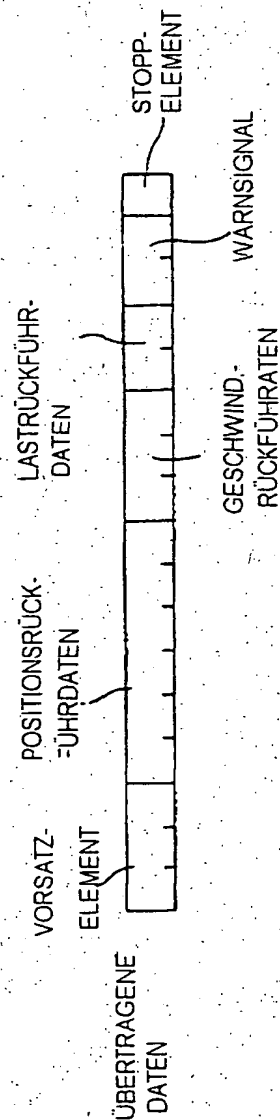
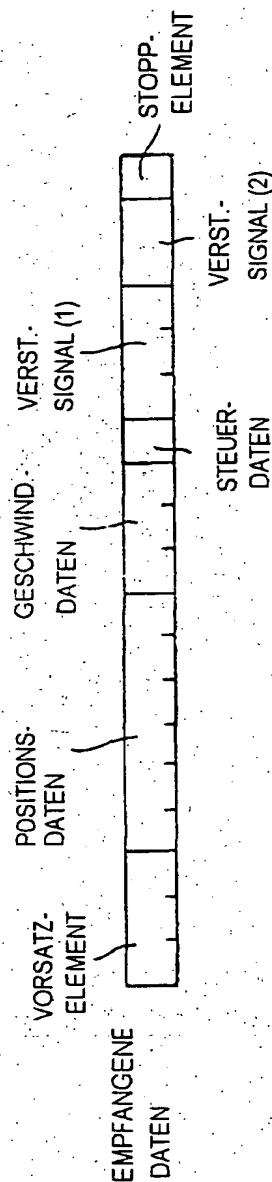


FIG. 14



7 10 9

FIG. 16

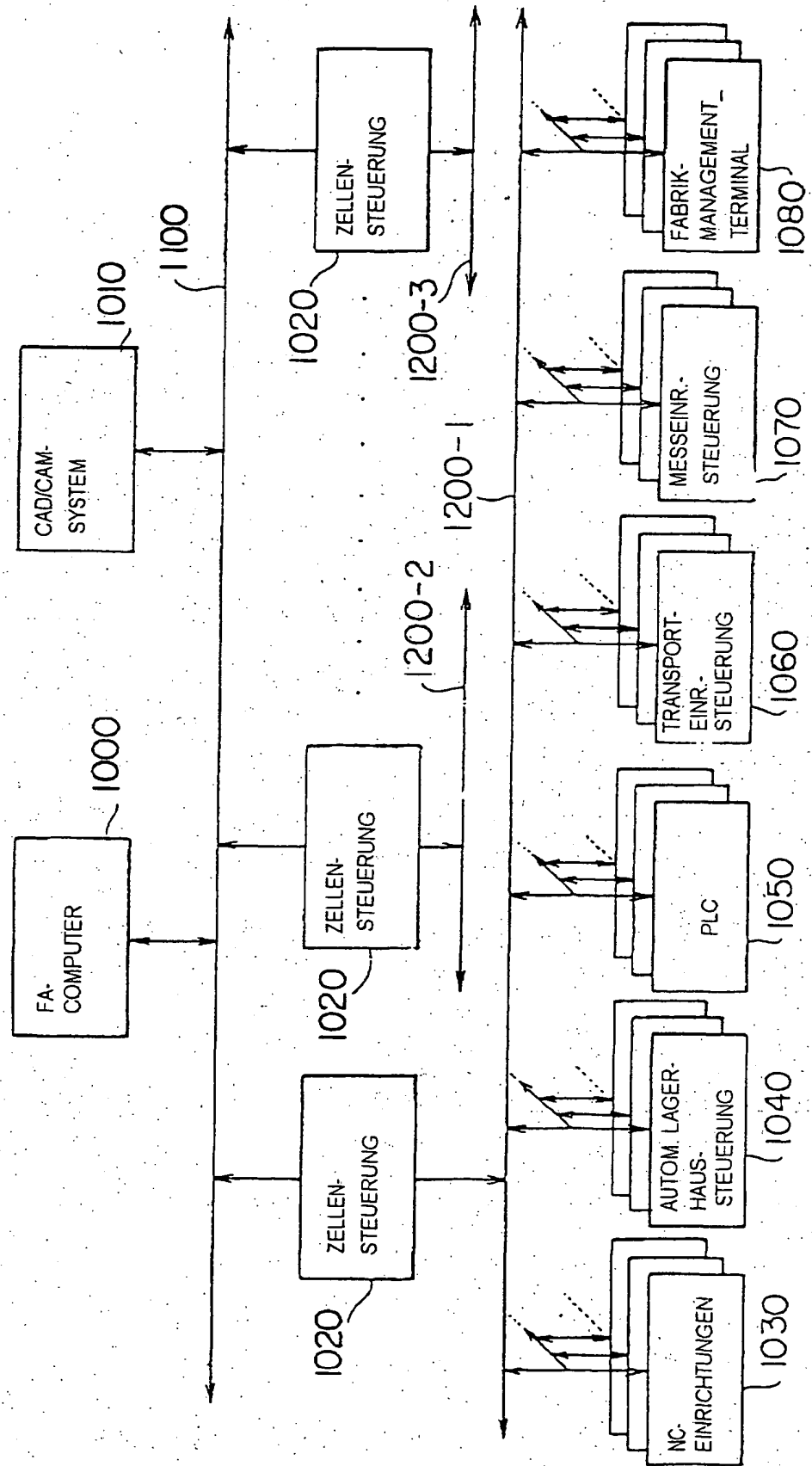
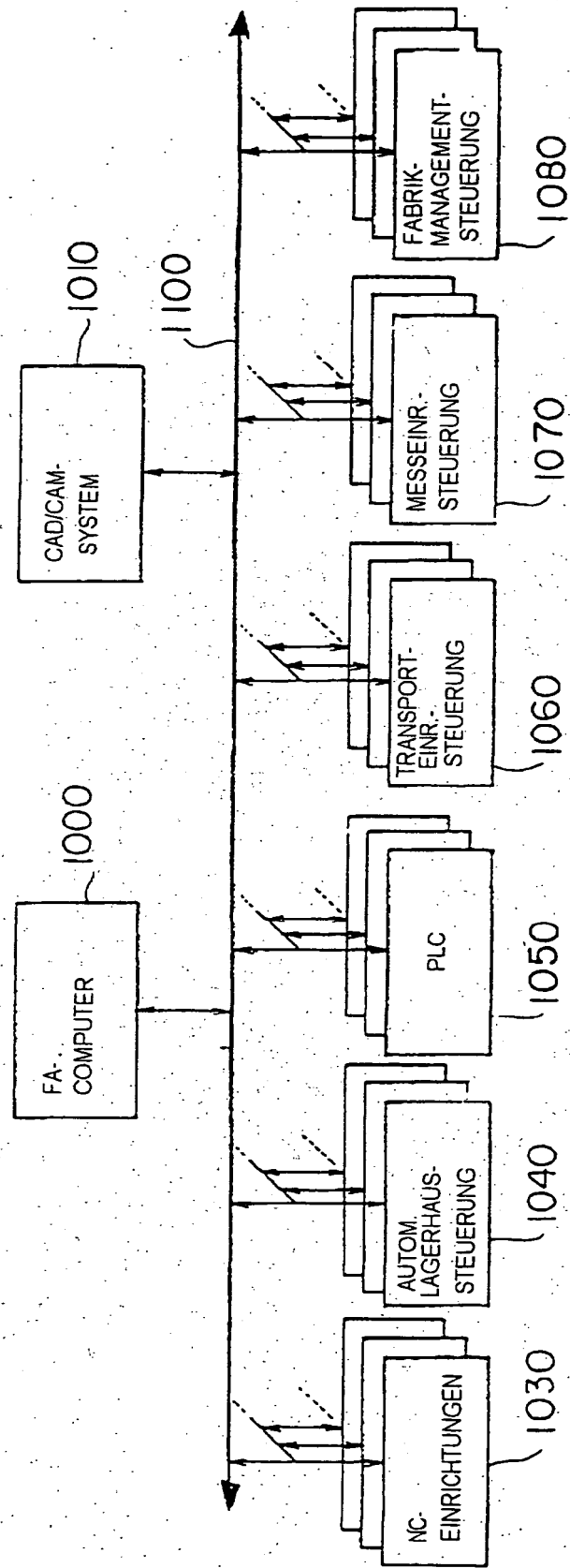
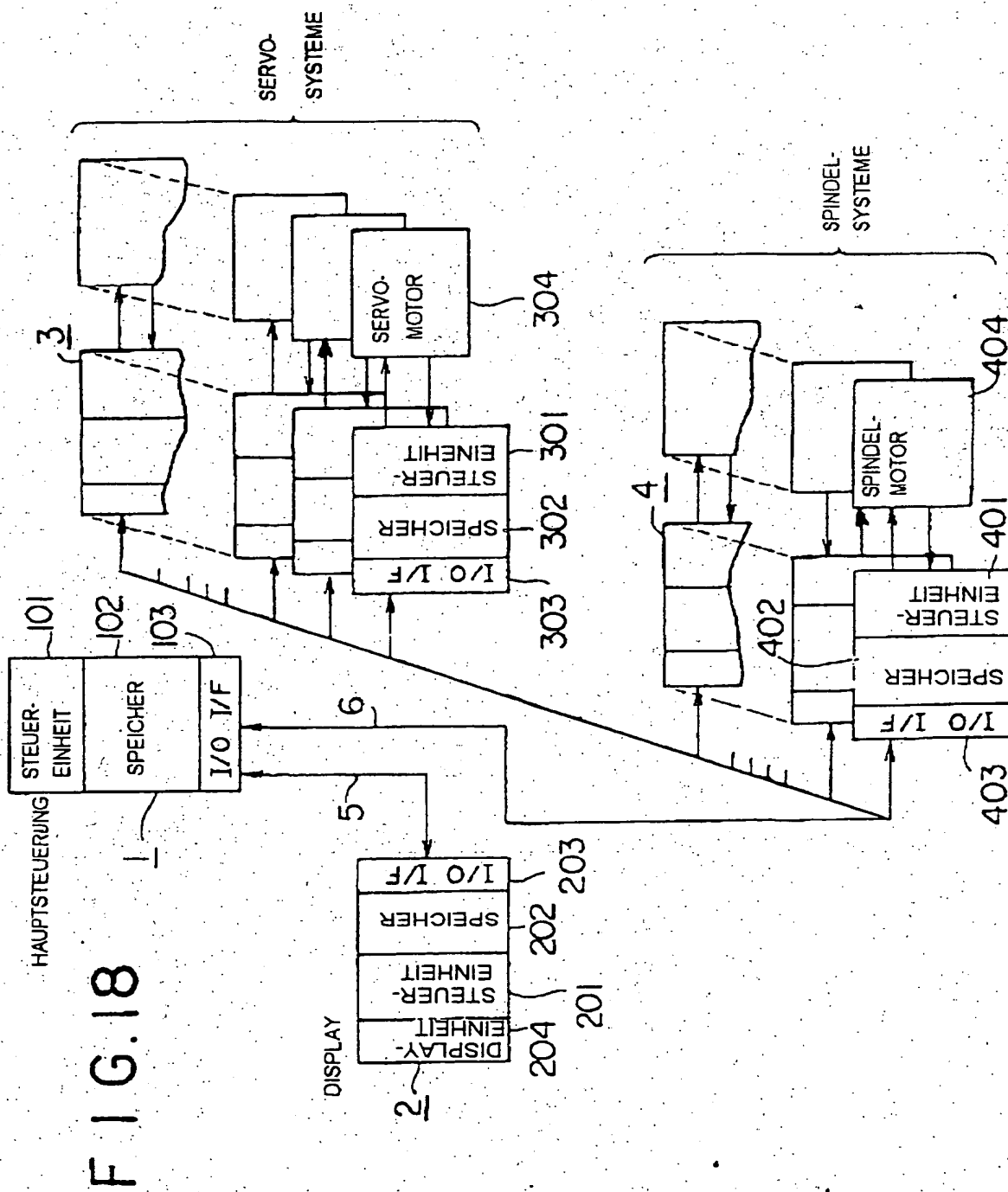


FIG. 17



17.10.87

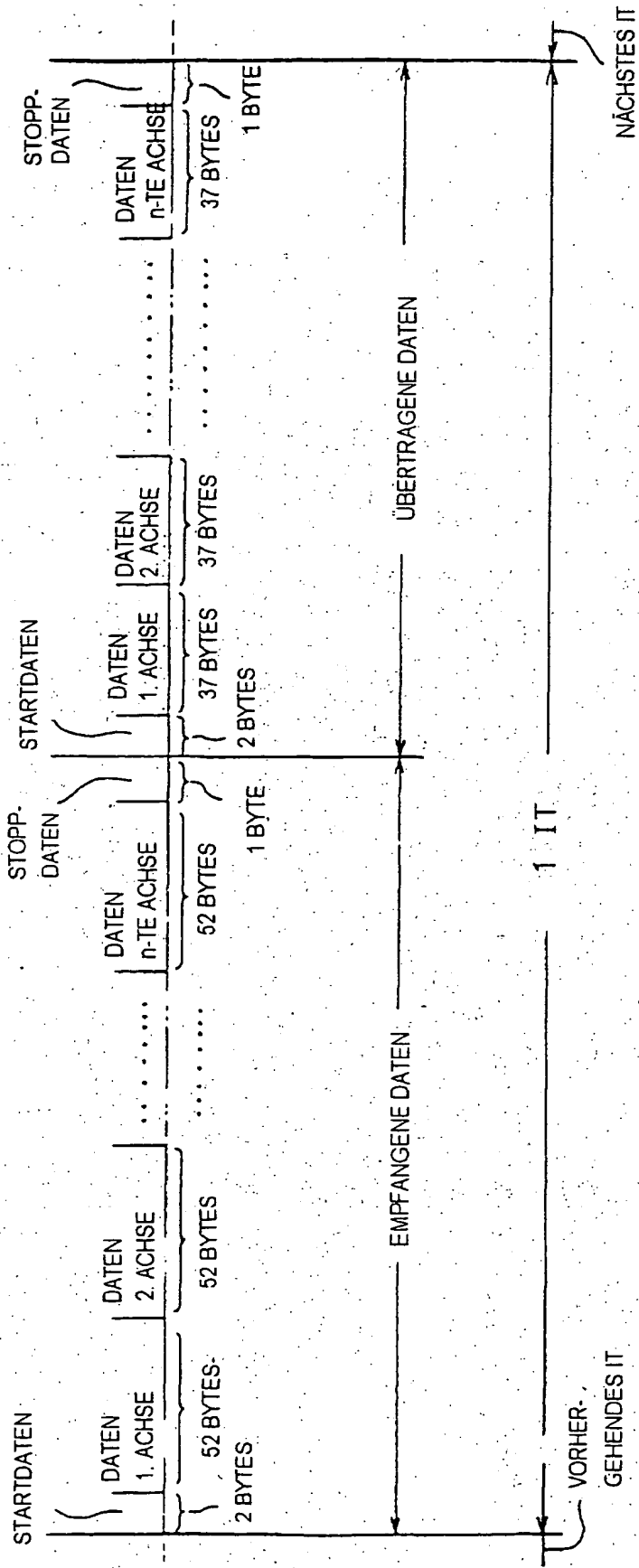


17.10.97

FIG. 19

	VORSATZ-ELEMENT	DATEN-ELEMENT	ENDELEMENT		LÄNGE IN BYTES	
(a) POSITIONS-DATEN	3	8	1		12	BYTES
(b) GESCHWIND.-DATEN	3	6	1		10	BYTES
(c) STEUERDATEN (1)	3	1	1		5	BYTES
(d) STEUERDATEN (2)	3	1	1		5	BYTES
(e) VERST.-SIGNAL (1)	5	5	1		11	BYTES
(f) VERST.-SIGNAL (2)	5	3	1		9	BYTES
				GESAMTBYTES VON EMPFANGENEN DATEN	52	BYTES
(g) POSITIONS-RÜCKFÜHR-SIGNAL	3	8	1	END-ELEMENT	12	BYTES
(h) GESCHWIND.-RÜCKFÜHR-SIGNAL	3	6	1		10	BYTES
(i) LASTRÜCK-FÜHR-SIGNAL	3	4	1		8	BYTES
(j) WARNSIGNAL	3	3	1		7	BYTES
				GESAMTBYTES VON ÜBERTRAGENEN DATEN	37	BYTES

FIG. 20



17 10 97

